Worcester Polytechnic Institute
Digital WPI

Interactive Qualifying Projects (All Years) | Interactive Qualifying Projects

March 2015

Predicting the Price of a Stock

Jacob Grotton
Worcester Polytechnic Institute

Sidney L. Veilleux
Worcester Polytechnic Institute

Predicting the Price of a Stock

An Interactive Qualifying Project Report

submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Jacob A. Grotton

Sidney L. Veilleux

Date: March 4, 2015

Project Advisor:

Mayer Humi, PhD

Table of Contents

Table of Equations

Table of Figures

Table of Tables

Abstract

The goal of this project was to apply research and signal processing techniques towards the development of an accurate model for the prices of securities traded on American stock exchanges, with the intent of producing short-term forecasts. The final model utilized previous stock prices, exchange index values, and company research, with the purpose of providing investors a tool for making informed financial decisions. This model was tested using several trials of virtual investment portfolios, encompassing a wide range of stocks.

Executive Summary

This Interactive Qualifying Project was completed over the course of three academic terms, of which the focus of each was divided into a different facet of stock price model development. The overarching purpose of the project was to create a model to predict the future prices of stocks traded on American exchanges. The development of such a model would greatly help unseasoned investors with very little knowledge of the workings of the stock market, allowing them to earn a profit without having to perform research of their own. In an age where it seems as if you must be a big investment bank or a hedge fund to make money trading stocks, the implementation of this model would allow individuals to have a better chance of generating income in this manner.

Over the course of Term A, the team used various signal analysis techniques to implement a mathematical model of the daily closing price of a stock. This model took into account no other inputs beyond previous values of the given stock's price. During this term, different combinations of analysis methods were experimented with, and those which provided the most accurate of forecasts were chosen for future research.

During the second term of the project, the efforts of the team were focused on the development of a model for American stock exchange indices, and its integration into the stock models of Term A. Several models were produced using the analysis methods developed in the previous term, and that which most accurately predicted future stock exchange values was chosen. Five equations were developed for combining the basic stock model with that of the exchange, and their respective constant coefficients were calculated using iterative computational analysis in MATLAB. Finally, the equation which produced the least forecast error over a wide range of stocks was chosen for future analysis. This new integrated model was put to practice on a virtual portfolio of stocks, using one month of forecasts to make investment decisions.

With a fully-developed mathematical model of the price of a stock and the value of the exchange upon which it is traded, the team utilized Term C to integrate qualitative company research and market analyst opinions into the overall model. This three-part model allows investors to make informed decisions, based on a given company's news presence, the opinions of popular financial advisors, and the mathematical model. During this term, the new model was tested extensively with several rounds of virtual investment portfolios, as well as research into emerging technology companies.

Introduction

The goal of this Interactive Qualifying Project is to produce a model which accurately predicts the future prices of stocks traded on American exchanges. This model, which was developed over the course of three academic terms, is based upon the daily closing price of a stock, which is viewed as a discrete, real-valued time signal. Accurately predicting the future values of stocks can allow one to make concrete investment decisions, minimizing the risk of capital loss. The creation of such a model can be very beneficial to new investors who may not know much about the stock market, but desire a method to easily earn income by trading stocks.

This reasoning behind the efforts of this IQP is simple: many people know about the earning potential of the stock market, but are unsure where to start or how to generate a profit. There exist many simple internet-based platforms that allow anyone to trade in stocks, but a new investor generally has no idea how to invest their hard-earned capital. This IQP team sought to apply signal analysis techniques learned as undergraduate electrical engineers towards the creation of a system that would give advice to small investors who have very little knowledge in the workings of securities exchange, but would like to earn a profit.

Over the course of the project, the team researched and implemented a variety of signal analysis techniques, which were used in the creation of several models. These models were tested on a variety of stocks, ranging widely in both value and sector of American industry. The forecasted stock price values produced by each model were compared to actual stock prices in order to determine their prediction accuracy. As the project progressed, the models became more complex, taking more factors into consideration in order to maximize this accuracy.

Background

The concept behind the modern stock market can be traced back to a simple system that began in 17th century Europe. In this era, wealthy business men would come together and help their favorite merchants by buying goods and supplies for their travels, thus “taking stock” in these agents. In return for the investments, the merchants would repay their benefactors with either better deals on their goods, or with a share of the profits, similar to modern-day dividends. This idea was thought to be mutually beneficial, providing the trader could be trusted not to go out of business. Ideally, profits would increase for both parties: the merchant would acquire more trading resources, while the businessmen would gain increasingly more money, as this was a never-ending cycle.

In modern day, this basic concept still exists, yet on a much larger scale. Since there are far greater companies today than there were merchants in the 1600s, the entire stock market gets to share in the wealth of success. However, this means that it is harder to know if investments will come to fruition, as a modern company is much more complex than one merchant who can be trusted not to go bankrupt tomorrow. Therefore, trading in modern stock markets provides a much higher level of risk than in the past. With a higher level of risk comes a greater magnitude of possible profit, along with an equal level of exposure to possible loss. Modern technology has made the stock market a way for nearly anyone to make money just by having capital and knowing where to invest it. Therefore, it is widely used by people around the globe to invest their retirement savings and other capital into stocks, in hopes to turn a profit.

Since one is able to make money purely by having money to invest, trading in stocks has become increasingly popular. In addition, modern financial concepts allow an investor to make money from positive or negative changes in stock values. A stock can be either bought, with the hopes that its value will rise, or short-sold, based on the assumption that it will decrease in value and can be bought at a price lower than the amount at which the investor offered to sell it. Accordingly, if an investor were to know the future price of a stock, they could make money regardless of the outcome, providing the value does not remain constant. This property makes the stock market a very good candidate for modeling, as being able to accurately predict future values of the signal can result in the realization of profits.

Applying knowledge from signals analysis courses, the team was able to interpret the price of a stock as a real-valued signal that is discrete in both time and magnitude. Even though the price of an American stock is changing constantly between the hours of 9:30 and 4, the analysis methods used in this project sample this value at a constant rate of once per market day, the security's daily closing price. This signal is considered to be discrete in amplitude as well due to the fact that the monetary values used as inputs are quantized to whole dollars and cents. There exist a multitude of analysis techniques for modeling and forecasting signals of this nature, and several of them were applied to real historical closing stock price data over the course of the project.

Research

Autocorrelation

Autocorrelation is of key importance to short-term stock signal modeling. This process is used to find the amount of relevant discrete time signal values to apply to the models, as it correlates to the most recent data. For the application of short-term stock signal modeling, this is applied to one year of daily closing stock prices. Assuming the stock to be a complex periodic waveform, this is done to determine a mathematical estimate of the length of one period, along with eliminating some of the lower frequency components that appear in a relatively large data set. The autocorrelation value is found by shifting the data by an increasing number of samples, and finding the correlation between the original and shifted time series. The data is continuously shifted (lagged) until the correlation between the original and shifted series is equal to zero. At this point, the number of lags is equal to the autocorrelation period, and this number is used to determine how many samples to use for model creation and signal analysis. For an autocorrelation value of n, it is assumed that the n most recent samples of the series are relevant data in regards to the creation of short term forecasts. This number is also assumed to be the fundamental period of the stock signal. Autocorrelation is applied to every dataset of stock prices to determine this period prior to each of the signal modelling techniques produced by the team.

Least-Squares Linear Regression

Least-Squares is a commonly used method for calculating the linear regression of a set of data. This is generally done by predicting a line for the set of points and then calculating the average square distance from that line to each point. This is to make sure that the line properly represents the entire data series and not just parts of it. The process is repeated until the line with the lowest average square distance from the data is found, thus producing the line with the least error. The slope and y-intercept of this line is used to define the Least-Squares Linear Regression for the set of data points. In the context of this project, this is used to determine the most basic linear trend of a stock signal, which serves as the base for many of the models created. This trend often represents the most basic direction towards which a stock is headed, and often correlates to the overall trend of success of the company.

Fourier Series

The Fourier Series is a method in general time signal analysis that is used to model data using a sum of sine and cosine waveforms. The number of sinusoidal terms used to create the series is defined by the order of the Fourier Series. For example, a Second-Order Fourier Series, which shall be referred to henceforth as FourierTwo, contains two sine and two cosine terms. The more sinusoids that are used, the more accurate the dataset that is given will be modeled, for past value interpolation. However, using higher order Fourier Series generally results in less accurate forecasts, as any future data points (outside the original range of the time series) are obscured by a relatively large spike in magnitude, as a result of Gibbs Phenomenon. This phenomenon states that periods of many added sinusoids will eventually align and cause for a large magnitude spike in data. This is observed in the application of this project, due to the fact that the forecast values lie at the edge of the dataset, which happens to be (by definition) one fundamental signal period in length. For the purposes of this project, no more than six sinusoidal terms are used, to prevent this phenomenon from obscuring the accuracy of forecast results. Therefore, Fourier Series of order three or fewer were used for analysis.

Generally, the amplitudes and frequencies of these sinusoidal terms are found by converting the series into its Fourier Transform representation in the frequency domain. In the frequency spectrum, only the highest magnitude terms are kept, while the others are discarded. The number of terms to be kept is defined by the order of the series. This yields a signal with much less noise, since only a few fundamental sinusoids remain, while still following the general shape of the original signal.

ARIMA

The Autoregressive Integrated Moving Average (ARIMA) model is a common time series analysis technique developed by statisticians George Box and Gwilym Jenkins which combines three types of time-series analysis techniques, in an attempt to model a set of data and use this model to forecast future data points. The first analysis technique implemented by ARIMA is Autoregression. An autoregressive model is essentially a linear regression, based on the assumption that current and future values of a time series depend on the sum of a constant intercept, a random noise value, and previous series values multiplied by coefficients. An autoregressive model of order p (denoted AR(p)) contains the sum of p previous terms of the series multiplied by p coefficients, as defined in Equation One.

$$x(t) = \xi + \varepsilon + \phi_1 x(t-1) + \phi_2 x(t-2) + \ldots + \phi_p x(t-p)$$

Equation 1: Autoregression

Another major component of the ARIMA model is Moving Averages analysis. This technique operates under the assumption that each element in the series can be affected by past error values which cannot be accounted for by Autoregression. This type of model, of order q, is defined as a constant μ (the mean) summed with q previous errors, multiplied by moving average coefficients θ, as can be seen in Equation Two.

$$x(t) = \mu + \varepsilon_t - \theta_1 \varepsilon(t-1) - \theta_2 \varepsilon(t-2) - \ldots - \theta_q \varepsilon(t-q)$$

Equation 2: Moving Averages

The third component of the ARIMA model is Integration. This analysis technique is based on the average change between past periods, assuming that this change will continue. In this model, present values depend on the corresponding values in past periods, added to a constant term α representing average change between periods. An I(1) Random Walk model follows the calculation outlined in Equation Three.

$$Y(t) = Y(t-1) + \alpha$$

Equation 3: Integration of Order 1

Volatility

Another metric of time series data analysis is volatility, which was integrated into the stock price prediction model. Volatility is a measure of how much a signal is changing, and can be measured by calculating the standard deviation of stock price data over the course of one autocorrelation period. Standard deviation, denoted σ, is the square root of the sum of differences between each term in the series and the mean, as defined in Equation Four.

$$\sigma = \sqrt{\frac{\Sigma(x-\bar{x})^2}{n}}$$

Equation 4: Volatility (Standard Deviation)

In statistics, volatility is a measure of the deviation of a series from its mean. From a financial viewpoint, volatility can be used to determine how much the value of a stock is changing, which can be indicative of the relative frequency at which the commodity is traded. Stocks with high volatility have a higher risk of changing unpredictably, while stocks with low volatility are relatively unchanging, and are considered to be less likely to diverge far from their mean in the near future. A high-volatility stock can yield very large returns, with the possibility of equally large losses. A low-volatility stock may not earn an investor as much gains, but the possibility of capital loss is considerably lower. Based on this knowledge, volatility was used to determine how much trust to place in the prediction models for each stock.

Using standard deviation to measure volatility results in each stock its own unique level of volatility, since this value is larger for signals of greater magnitude. With the goal in mind of using volatility as part of a universal stock model, this value was normalized to create a new metric to measure a stock's volatility. By dividing the standard deviation of a stock by its mean, a more generalized volatility measure was created, allowing the relative differences in volatility among stocks of different magnitudes to more accurately be compared. The mathematical definition for normalized volatility can be seen in Equation 5, where $\bar{x}$ denotes the signal mean.

$$V_n = \frac{\sigma}{\bar{x}}$$

Equation 5: Normalized Volatility

Volume

All traded stocks have a volume that indicates how many shares outstanding have been purchased by the investing community. This measurement is usually displayed as shares traded per day, which can also be used as an indicator of the popularity of a stock. Stocks with a high popularity are traded more often, therefore their volumes are much higher. Accordingly, the converse of this statement is true for less popular stocks.

The volume of a given stock can show much about how a given stock's price behaves in the context of long-term investing. High volume stocks generally act as safe long-term investments that produce reliable, consistent returns. These stocks are generally considered to be very low-risk by investors. On the other hand, low volume stocks tend to provide a higher level of risk, but with a higher possibility of paying off. Since low-volume stocks are less often traded, if the amount of people that invest in that stock increases even slightly the price of that stock increases accordingly as there exists an increased demand for a commodity that is limited in quantity.

Exchange Indexes

Exchange index values were also used as a component of the stock price model. An exchange index, or stock market index, is a metric used to estimate the value of a stock market as a whole, which changes as stocks rise and fall in aggregate. This index is generally formed by the sum of prices of several blue-chip (high-value) stocks traded on an exchange, usually in a weighted average, to account for variations in the order of magnitude of prices among the stocks. The day-to-day changes of a stock market index can give insight into how the market is performing, which can be indicative of businesses' and shareholders' reactions to the current state of the economy. Often, the price of a stock will change in a similar fashion to the index of the exchange upon which it is traded. For example, a large rise or fall in the value of the stock market will be accompanied by a similar rise or fall in a large portion of the stocks traded within that market. Thus, changes in exchange indexes can be used to help predict changes in the price of a stock.

Methods

Linear Regression and Fourier Series Analysis

For the first model, least-squares linear regression was performed on the daily closing price of one autocorrelation period for each stock. The regression line was subtracted from the actual stock values, and these residuals were used as an input to second-order Fourier Series analysis (FourierTwo). The remaining residuals of the Fourier series were assumed to represent a low-magnitude noise in the stock signal, and the average of the minimum and maximum of this signal was used to determine a range of relative accuracy for the predictions. This range was used to create an error band for future stock price forecasts, and is present in each of the different models developed over the course of this project. The Fourier series was added to the linear regression line, and this was used as a model for future price predictions. Figure one shows the Linear Regression-FourierTwo model with upper and lower error bands in practice, used to predict one month of stock prices for NVIDIA Corp (NVDA, NASDAQ).

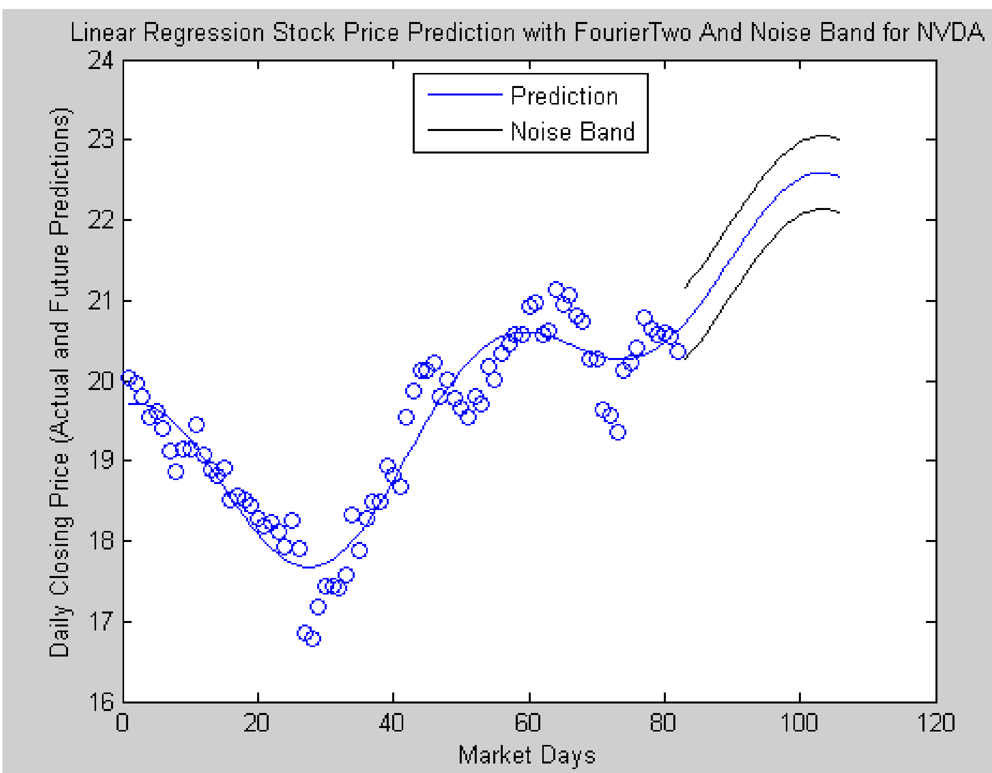

Figure 1: Linear Regression-FourerTwo Model

ARIMA

The second method of modeling a stock price made use of ARIMA analysis. Since ARIMA is defined in order by the three parameters p, d, and q, it became necessary to determine the best order of Autoregression (p), Integration (d), and Moving Average (q) calculations for this particular application. In order to accomplish this, a MATLAB script was written to perform stock price modeling for twenty stocks, providing one month of forecasted prices (20 market days), starting one month in the past. This script iterated over every combination of p, d, and q parameters from ARIMA(0,0,0) to ARIMA(3,3,3), as orders higher than three are rarely used in the application of short-term forecasting. For each ARIMA model, the maximum and average forecast error was stored, as the forecasting time period contained known values that could be compared to predictions. After several hours of computation, the iterations revealed that ARIMA(2,1,2) provided the most accurate predictions of stock prices for one month of analysis. The MATLAB function *estimate* was used to calculate the coefficients of each term in the model, and *infer* and *forecast* were used to infer past values and to predict future values, respectively. Figure two depicts an ARIMA(2,1,2) model for Intel (INTC, NASDAQ).

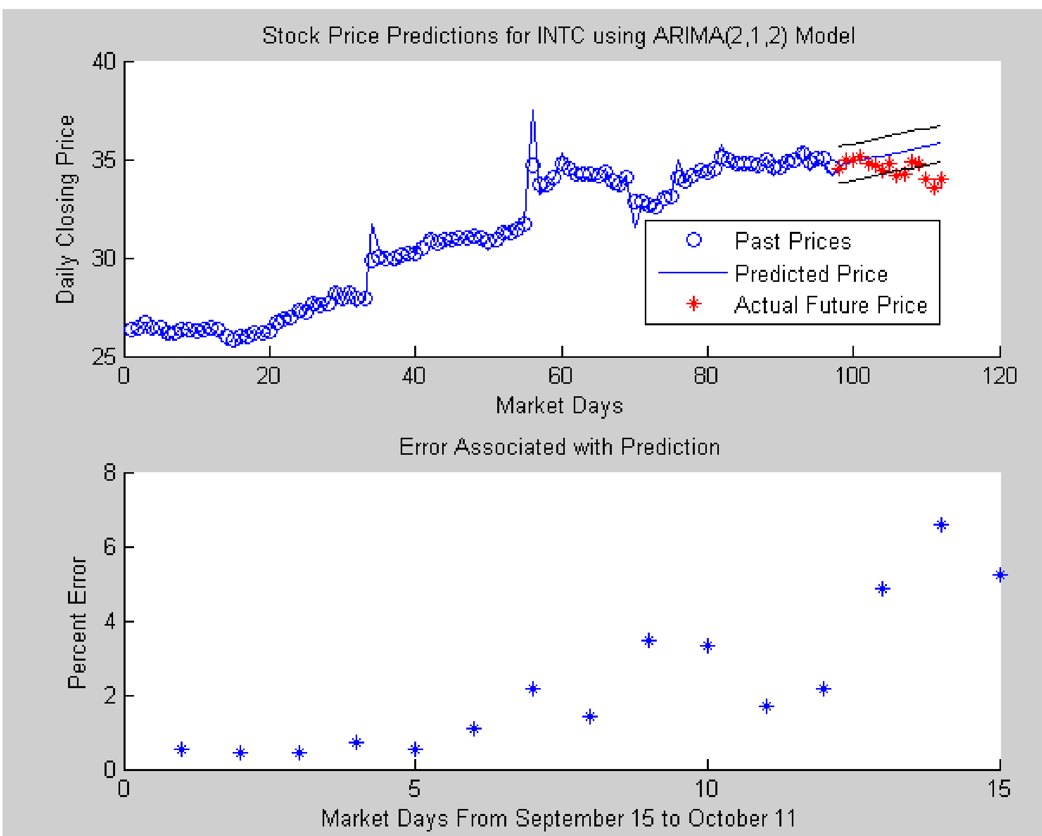

Figure 2: ARIMA(2,1,2) model of Intel

Fourier and ARIMA

Over the course of the analysis performed in Term A, it was observed that the ARIMA(2,1,2) model predictions were accurate compared to the actual future stock prices, but did not continue to follow the sinusoidal component present in many of the time series, which resulted in relatively linear predictions. This caused the accuracy of these models to quickly diverge after less than one week of forecasts. In an attempt to correct this issue, Fourier series analysis was performed on the original stock price time series, and subtracted from the series to produce residual values. ARIMA(2,1,2) analysis was then performed on these residuals. The second-order Fourier series was added to the ARIMA model to produce a new model combining the two techniques. This ARIMA-FourierTwo model was used to produce one month of predictions for a given stock. An example of the application of this method of analysis can be seen in Figure Three, which provides an interpolation model of one autocorrelation period of Yahoo (YHOO, NASDAQ).

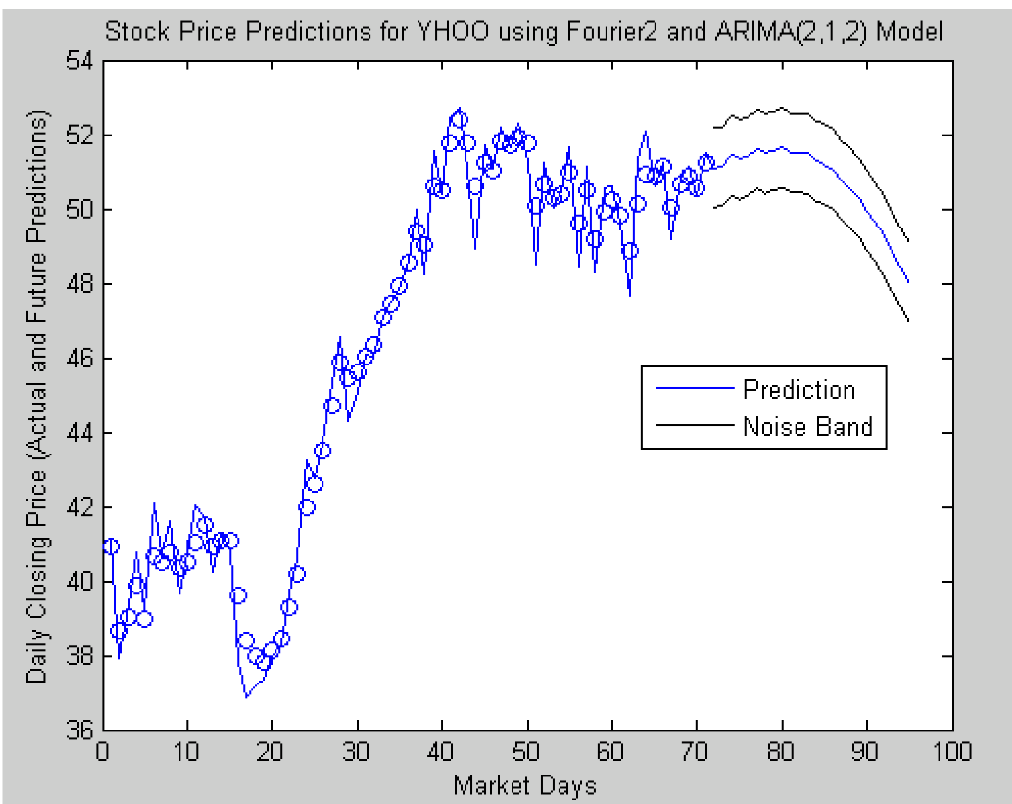

Figure 3: FourierTwo-ARIMA Model

Term A Models

During the final weeks of Term A, six models were produced to try to accurately predict one month of values for the stocks. These six were created using twenty stocks from the Technology and Food Service market sectors, and were tested on ten other stocks in the Banking and Finance industry. Some of the methods that were used are the same as listed above: Linear Regression with FourierTwo series analysis and ARIMA(2,1,2) analysis. The other four are slight variations on previous methods, to add to a wide variation of models in order to potentially find a successful model type.

For the first of these experimental models, the order of the model was tested. Firstly, a second order Fourier series analysis was taken over the autocorrelation period for each stock. The difference between this model and the actual stock was then taken and a linear regression line was fit to the remainder. The difference between this linear trend and the remainder is assumed to be natural noise in the model and kept to create an error band for the model. Finally, the FourierTwo and the Linear Regression models were added together. This result had a fair chance to be different from its Linear-FourierTwo counterpart, depending on how volatile the stocks were. It was observed that lower volatility stocks had a less chance of being modeled different by Linear-FourierTwo compared to FourierTwo-Linear.

Next, a slight variation of the previously mentioned FourierTwo-ARIMA(2,1,2), was tested. This model is very similar in that it combines some of the different methods that were used in previous models. However, this model is a combination of Linear Regression and ARIMA (Linear Regression-ARIMA(2,1,2)). This is produced in a very similar way to most models. Firstly, a linear representation of the stocks is produced. Then, the difference is taken between the actual stock and the regression line. Using the residual values of this subtraction, an ARIMA(2,1,2) model is produced. The ARIMA model and the linear regression are summed, creating a model that can be used to forecast future stock prices. The other Term A models made use of volatility, volume, and stock exchange index values, which are discussed in detail in the next sections.

Volatility Based Model

Over the course of Term A, the Linear Regression and FourierTwo model, as well as the FourierTwo and ARIMA(2,1,2) model, were applied to a variety of stocks that concluded in mixed results. It was observed that in general, signals with lower volatilities were best modeled by a linear trend combined with a sinusoidal component, while higher-volatility stocks were more accurately modeled using FourierTwo and ARIMA(2,1,2). It was further observed that for very low volatility stocks, the sinusoidal component often over or under-predicted the actual peaks and dips of the signal, respectively. Based on these observations, a new model was created using normalized volatility as a metric for deciding which analysis technique to use for interpolating past values and forecasting future prices of a given stock.

Stocks which were considered to be low volatility, with a normalized volatility of less than 0.35, were modeled using Least-Squares Linear Regression, performing second-order Fourier Series expansion on the residuals of the trend line, and scaling the amplitude of the sinusoidal component by a factor of 0.4 before adding the it to the trend. Mid-volatility stocks, with normalized volatilities between 0.35 and 0.8, were modeled using the same technique, but without the amplitude scale-factor on the Fourier Series component. High volatility stocks, characterized by having normalized volatilities above 0.8, were modeled using FourierTwo, combined with ARIMA(2,1,2)analysis performed on the differences between the Fourier series and the actual time series. The resulting model was implemented in MATLAB using *if-else* statements to select the appropriate model for a given stock based on the value of its calculated normalized volatility.

Exchange Model

In the search to improve upon the models, it was noted that the exchange indices acted very similarly to their relative stocks. It was then thought that it could be easier to attempt to model the exchange index and then apply it to the stock models. This was to be added as an element to decrease total error for the model. The task of modeling the exchange index was treated just like all the other stocks modeled over the course of the project. For the majority of exchange index models, a Linear-FourierTwo model was used to represent the series' given values. This was chosen after each of the models that have been tried for stocks were applied to the exchange index. Due to the low volatility of the New York Stock Exchange and NASDAQ indices, this method of signal modeling was used extensively. The main goal of this was to keep the model very simple to avoid the addition of any error. Any decreased level of prediction accuracy would propagate into stock value inaccuracy when the models are combined. Much of the effort towards model development in Term B was spent attempting to combine the base stock models with the exchange index model in order to produce accurate forecasts. Several different methods of model combination were implemented, each based on different stock attributes, outlined below.

Volatility Application

One combination of stock and exchange models was designed based upon the observation that for many of the stocks, signals with relatively high volatility seemed to follow the movements of the low-volatility exchange indexes for the markets upon which they are traded, while lower volatility stocks were much less influenced by the movements of the exchange. With this observation in mind, an equation was written to combine the two models using weighted addition, with volatility as a metric to determine how much faith to place in each model. This was accomplished using the ratio of the volatility of the exchange index compared to the volatility of the stock.

First, the volatilities of the exchange index and stock price were evaluated for one period using Equation Four. These values were normalized as defined by Equation Six, by dividing the volatility of each time series by its mean, producing $V_{ne}$and V_{ns}, normalized volatilities of the exchange index and stock, respectively. Since a time series with higher volatility is generally less predictable, with a higher error of forecast results, the series with a higher proportional volatility was given a lower proportional influence in the results of the model. Based on the assumption that a lower volatility stock is easier to model without the need of exchange index influence, the ratios in equation five were used to create a new model using the weighted addition of the stock price model and exchange index model, where X_e is the exchange index model and X_s is one of the three stock price models.

$$x(t) = \frac{V_{ne}}{V_{ne} + V_{ns}} X_s + \frac{V_{ns}}{V_{ne} + V_{ns}} X_e$$

Equation 6: Weighted Combination of Models

Potency Application

Over the course of Term B, several other methods for combining the stock and exchange models were also implemented, using a variety of equations based on different observations made. Many of these methods made use of a scalar term α, which defined the potency of the exchange index model in regards to its weight in the overall stock model. A larger α corresponds to the exchange model having a greater influence on the movement of a stock, while smaller α values imply that the exchange index has very little potency in regards to a stock's future price. Some of the implemented model combination methods also made use of another constant, β, which represented the potency of the base stock signal model in regards to the overall model of the stock's price.

Determination of the values of these scalars, which were believed to be universal factors to be applied for all stocks, was performed using experimentation and model iteration in MATLAB. The script functions that implemented the models were modified to accept alpha and beta as input arguments, and to return the average and maximum forecast errors across all stocks. Another script was written to iterate over hundreds of values for alpha and beta, storing the error values in an array, which was finally plotted in three dimensions, with x and y representing alpha and beta, respectively, and z representing the error associated with each pair of α,β values. The minimum value in these plots signified the pair of alpha and beta scalar values which produced the lowest prediction error, and these values were chosen for analysis with a separate set of stocks for further testing.

The first of these methods implemented the use of α in the simplest manner, scaling the exchange model by α and adding it to the stock model scaled by (1-α), as defined in Equation Seven. Using this implementation, the sum of the amplitude scale factors always sum to one, resulting in a model which theoretically creates a discrete waveform of the same amplitude as the stock signal.

$$x(t) = (1 - \alpha)X_s + \alpha X_e$$

Equation 7: Complimentary Addition with Alpha Scale Factor

Another method for model combination was implemented using the Humi Equation, which relies on the assumption that the price of a stock can be predicted using the sum of a basic stock model and an exchange model multiplied by the correlation between the two signals, and scaled by a factor of α. In this manner, the value of α could be negative, to compensate for a negative correlation between the stock and exchange signals. This model combination method is described mathematically in Equation Eight.

$$Y = X_s(1 + \alpha * r_{SE} * X_E)$$

Equation 8: Humi Equation for Model Combination

A third method for combining stock and exchange models was developed in week six of Term B. This method is similar to the Humi Equation, but separates the addition terms such that the stock model is not multiplied by the exchange model. This model operates under the conditions that the stock model must be scaled by a universal constant β, while the exchange model is scaled by α and the correlation between the signals. This method was implemented as described Equation Nine.

$$y[n] = \beta X_s[n] + \alpha r_{SE} X_e[n]$$

Equation 9: Scaled Addition using Alpha and Beta

Each of these equations for model combination were tested extensively in MATLAB, performing iterative analysis to determine the constants that produce the most accurate results. After iterating over hundreds of possible values of each constant, yielding thousands of combinations for each equation, one equation and one set of coefficients were chosen for inclusion in the model. The selected method implements Equation Eight, with α equal to 0.1. Figure four shows predictions and actual values (shown in red) along with forecast percent error for one month of closing prices of Ebay (EBAY, NASDAQ) using the Linear Regression-FourierTwo model. At the right-hand side of the figure is a Linear Regression-FourierTwo model of the NASDAQ exchange index.

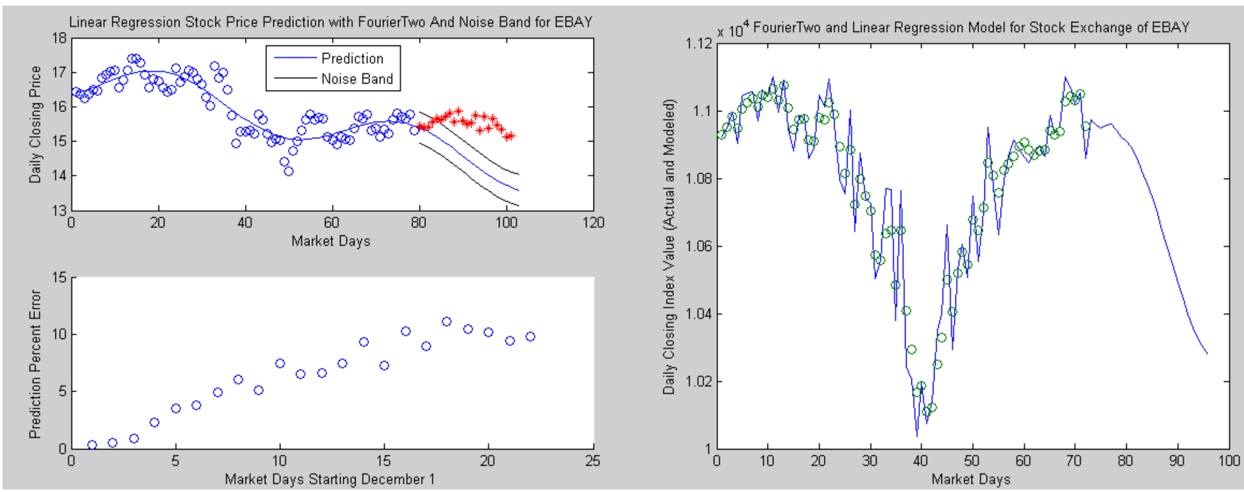

Figure 4: Linear-FourierTwo model for EBAY and corresponding index

As can be seen in Figure Four, the model predictions quickly diverge from the actual stock price within one week of forecasts. Figure five depicts the same stock, over the same period of time, using Equation Eight to integrate the NASDAQ exchange model into the predictions. This combination results in a slight shift in amplitude that, in this case, increases the overall accuracy of the model.

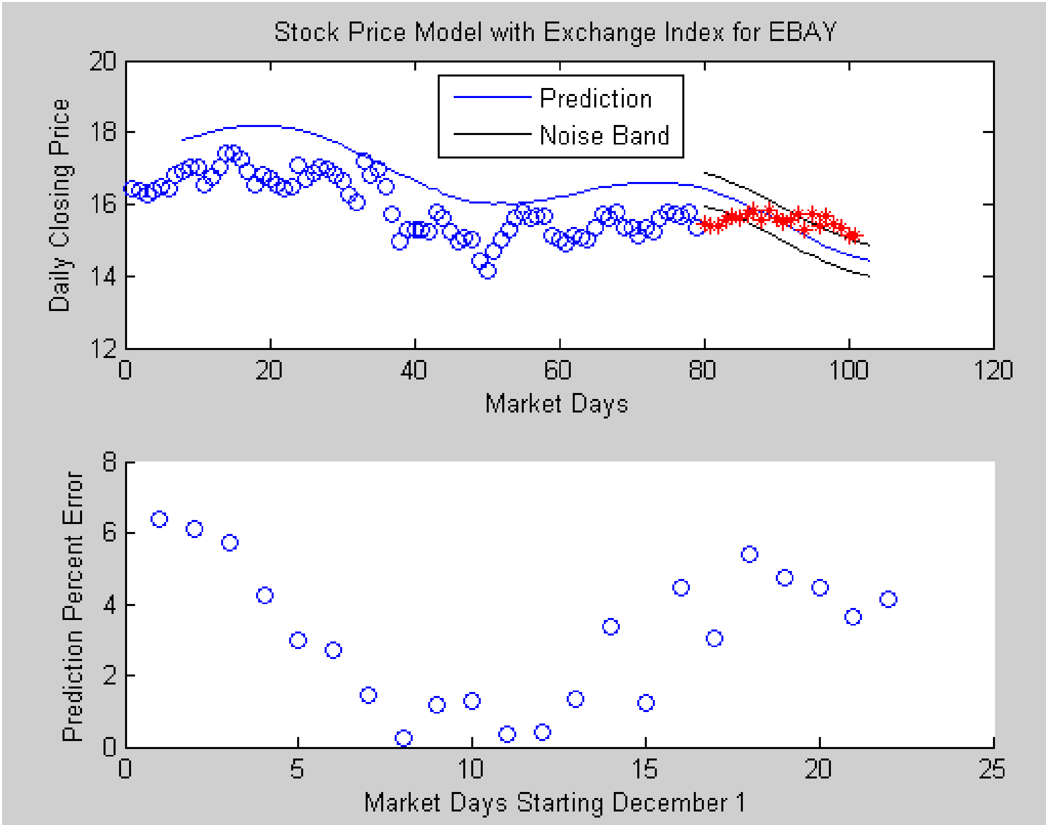

Figure 5: EBAY model with Exchange Index

Volume Application

Another metric applied to the modeling process for a given stock is its volume. Research was performed into seeing whether a correlation existed between the volume of a given stock and the accuracy of the stock's model. From a large sample of stocks and model results, the volume and the associated error were determined and plotted to see what type of correlation existed between these two metrics. The resulting plot in Figure six was produced.

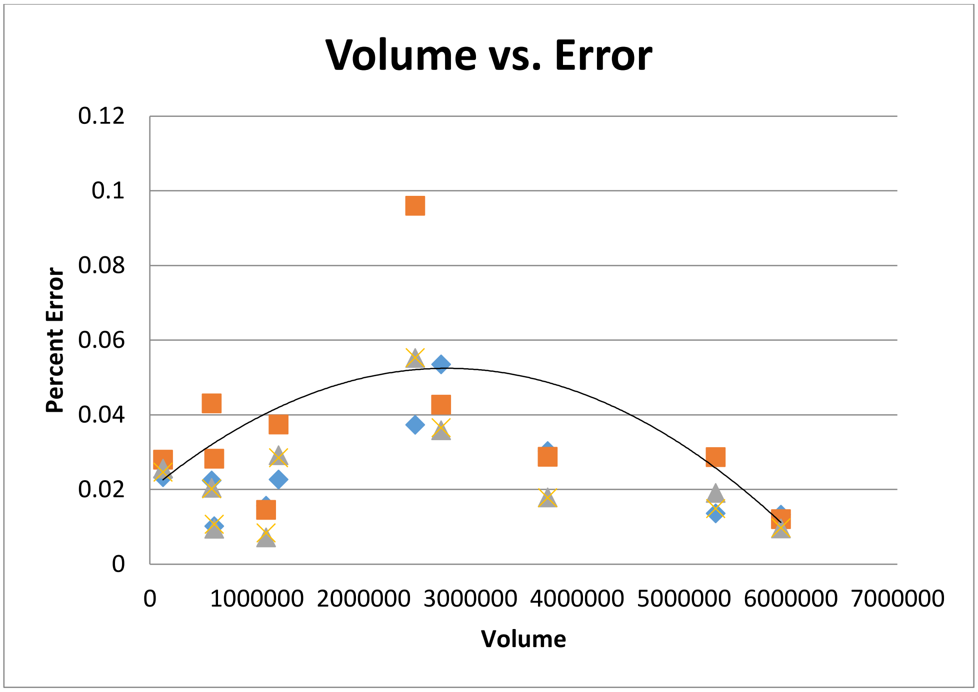

Figure 6: Volume Vs Error Plot

Overall, the magnitude of the error associated with model predictions is relatively low. The easiest stocks to model were the stocks that were traded the most. This is because stocks that are high risk are traded more often since investment with higher levels of risk can result in greater profits. On the other hand, many investors also prefer lower risk securities, especially for long term goals (such as retirement), as this ensures that the investments will eventually come to fruition. This made low-volume stocks easier to model as well. This left mid-level volume stocks as the hardest to model, as can be seen by the higher level of error for these stocks in Figure Six.

As a result from these conclusions it was found that the stocks with the most error should rely the most on the exchange models that were developed, with high and low-volume stocks relying more on their fundamental base models. The correlation value came directly from the plot, and was scaled down and normalized for modeling. The correlation was determined to be as follows:

$$\alpha = -3.2 * 10^{-14} * (\mathrm{Vol} - 3.0 * 10^{6})^{2} + .3$$

Equation 10: Factor for Volume

The relationship defined in Equation Ten is used to calculate the potency of the exchange. This result is then used in the same way as Equation Seven to combine the two models.

Figure Seven shows the basic FourierTwo model alongside the ARIMA models that are used together to make a decision. This was implemented prior to the use of the volume within the model.

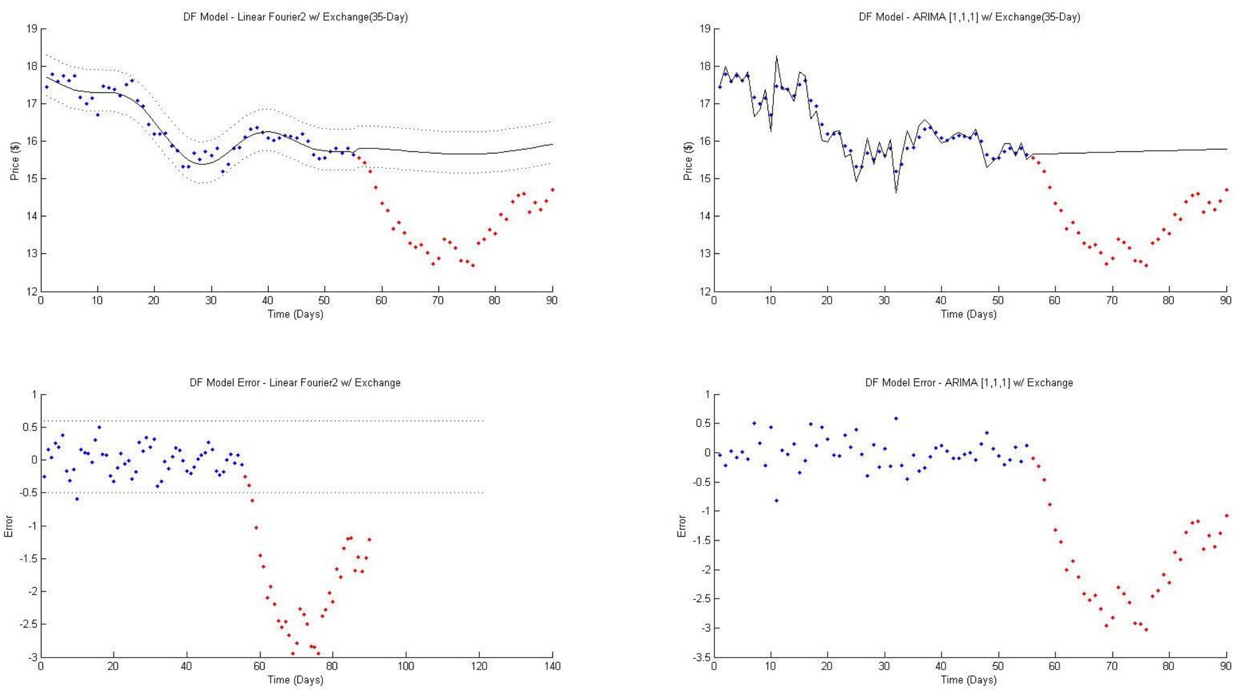

Figure 7: DF model without using Volume Equation

Figure Eight depicts the same model of the same stock in Figure Seven, taking into consideration the security's volume using Equation Ten. This was applied for both the Fourier and the ARIMA modeling techniques, combined with the exchange model. Figure Eight provides an example of how implementation of the volume greatly decreased the error of the model, correctly predicting a fall and subsequent rise in price.

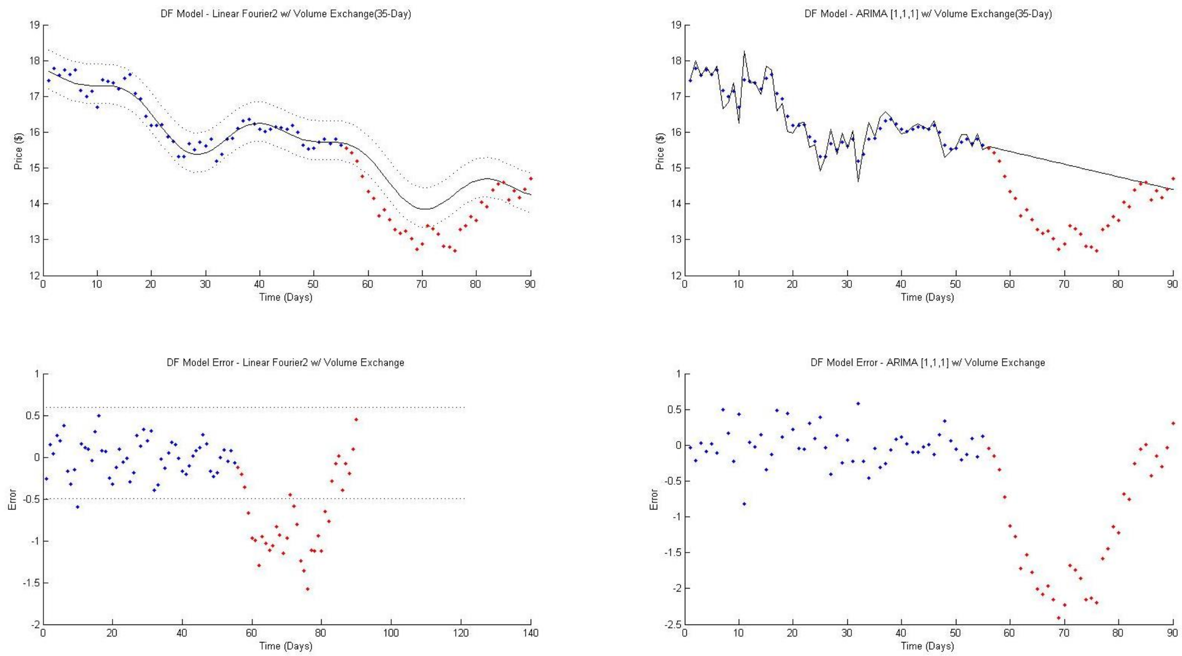

Figure 8: DF model with implementation of Volume Equation

Time Shift

Further inspection of the relationship between the change in price of a stock and the change in value of the index for the exchange upon which the stock is traded revealed another factor which was previously not included in the model combination. It was observed that the exchange index and stock signals made very similar movements, but the exchange often seemed to lead the stock by several days. For example, a rise in the exchange index value would be followed by a rise in many of the stock on that exchange several days later. This lead to the development of a new method for stock and exchange model combination that factored in this phase-delay between the two signals. The two models were added together using similar alpha constant as in Equation Nine, but with the exchange model delayed by a time shift factor *k.* This relationship is described mathematically in Equation Eleven. Determination of this time shift value was performed using the same iterative analysis techniques as mentioned above, with experimentation performed on varying values of alpha and k.

$$y[n] = X_s[n] + \alpha X_e[n-k]$$

Equation 11: Scaled Addition with Time Shift

Confidence

One technique used as a more general test of accuracy for the models was the confidence test. This came as a result of the multiple models that were developed throughout the project. This is not a quantitative number, but rather a visual aid in determining the accuracy of the developed models. To determine a level of confidence as to whether a stock will rise or fall, the following procedure was completed. This began by computing forecasted stock values using each of the models developed by the team. Next, the results of every model were visually displayed in a grid to view whether they produced similar predictions. If the majority of the models produced similar outcomes, predicting the same basic future trend, a high level of confidence was established for these results. The more models that agreed with one another, the more confidence was placed in the predictions of these models. This confidence system is an accumulation of all the other models used. This model is mainly used for decision making, as it does not offer any numerical results.

For example, both the Linear-FourierTwo and FourierTwo-ARIMA models produced similar forecasts over the same time period for Sonic Corp (SONC, NASDAQ). Because both of these models provided equal predictions, a strong level of confidence was placed in purchasing this security. This model correlation is outlined graphically in Figure Nine.

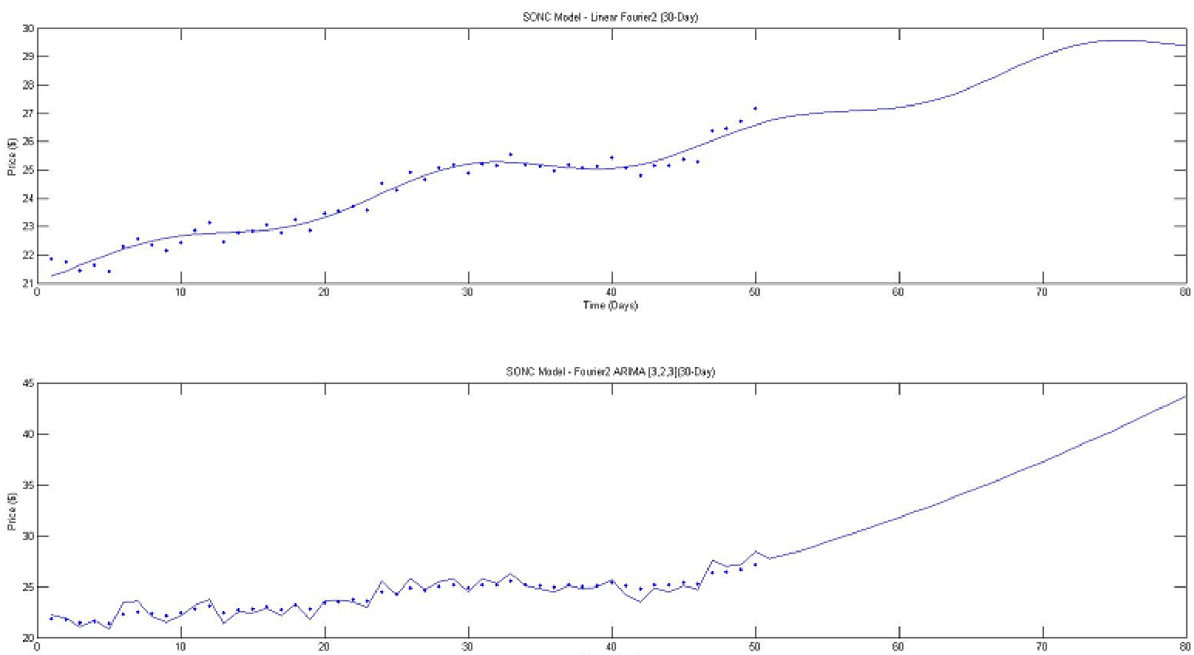

Figure 9: Summary of Confidence model for SONC

Research-Based Model

In Term C, a new model was developed, making use of not only historical stock prices and exchange indices to make estimates based on signal processing techniques, but also using researched facts about the companies which the stocks represent. To accomplish this, research was divided into two main areas: company news presence and market analyst opinion ratings. For a given stock, these two areas of research were applied along with the MATLAB-powered models produced in Terms A and B. If no two of these three analysis methods disagreed (that is, they all predict either increase or decrease in stock price), then the company was chosen as a viable source of income from either buying or short-selling its shares. If any two of the three sources contradict each other, then the stock is considered to be unpredictable for future values, and it is discarded from consideration. This is an expansion upon the model confidence developed in term B, in which a model is only trusted if all other models point to the same result. In this manner, an investor can be surer of the outcomes of trades, minimizing exposure to risk.

Company News Presence

A company chosen as a potential investment opportunity is researched extensively in the news. Using Google News Search, which provides an inclusively-wide spread of all available internet news outlets, a company is searched both by name and by stock ticker symbol. News articles are reviewed in a case-by-case basis, searching for positive and negative news stories. A company whose news presence contains any of the following topics is given a positive rating, as it is assumed that its value is soon to increase: expansion in size, buying new equipment or factories, new product releases, high quarterly earnings, new/renewal of business contracts, or a competitor goes out of business. A company whose news presence contains any of the following topics is given a negative rating, as it is assumed that investors will lose faith in the value of the stock and it will fall: contraction, losing business, subject of a lawsuit, under legal investigation, change in leadership, reports poor quarterly earnings, declares bankruptcy. Some news articles can be considered to be neither positive nor negative in regards to the price of its stock, and these are ignored.

Particular focus was placed on the news results from financial news sources, such as Seeking Alpha and the Wall Street Journal. Not only do these types of news outlets feature more relevant data on the financial performance of a company (shedding light on its possible success or failure), but these publications are read by millions of investors. If an article depicts a company in a very negative or positive image, and the majority of the investing public reads that article and acts based on this information, the stock price will rise or fall accordingly. Due to this effect, it can be stated that financial news outlets with wide publication and readership can have an indirect but timely effect on the future price of a stock.

Market Analyst Opinion Ratings

Another form of research into a company that is performed is focused less on the performance of a company and more on financial ratings. Many free resources exist for the common investor to receive the opinion of professional market analysts in regard to the direction in which the value of a stock is heading. These analysts generally give their opinions in the form of a Buy or Sell rating, sometimes accompanied by a modifier such as a "Strong Buy", "Weak Sell", or "B+ Buy". The website MarketWatch, provides such a rating for nearly every publically traded company in America. TheStreet is another such website, which is an extension of a company run by Jim Cramer, the host of the financial advice television program Mad Money. The opinions provided by these websites were used to form a positive or negative rating to integrate into the overall stock model.

As stated above in reference to financial news publications having an effect on the price of stocks, these market analysts have a similar power. A large fraction of the investing public looks to these types of advice websites and television programs for guidance, and many do as they are told, trusting the professional advisors' opinions wholeheartedly. In this manner, if Jim Cramer tells a television audience of millions of Americans that a particular stock will rise in value, many of them will immediately go out and purchase that stock. Microeconomic theory tells us that such an aggregate increase in demand for a commodity with limited quantity will cause the value of that commodity to increase in value accordingly. In short, the prophesies of popular stock analysts are self-fulfilling: if you tell the investing public that a stock will go down, they all sell it out of fear and that same day its price drops 10%. Because of this phenomenon, the opinion of these analysts must be taken into account for the creation of the overall model, since they have the power to make a stock's value rise or fall.

Three-Part Model with Feedback

The three part model described above was tested extensively during Term C, and used to create a portfolio of seven stocks from different industry sectors valued at $100,000, with the intent of holding the stocks for one month, selling on the 30th day. After two weeks of sitting on the portfolio, it had fallen 3% in value. After researching into why the new model had performed so poorly, it was observed that just days after the trades were made, several new articles came into publication that contradicted the previous conclusions made about some of the stocks. In addition, the market analyst websites had changed their opinions on another stock. These observations lead to the alteration of the model into one that incorporates feedback to decrease error.

The new model, which still features the three stages of research and analysis as described above, makes use of daily feedback at each stage. Each day, the process is repeated for every stock in the portfolio. The company is searched on Google News for new articles regarding its daily success or failure, the market analysts' opinions are taken into consideration, and the MATLAB model is run again with one new data point: yesterday's closing price. As students with a background in engineering, the team is very aware of the great value that feedback can provide to a system. In engineering, systems with feedback allow for quick response to impulse changes to a system input, and can greatly reduce the error at its output. This system is outlined graphically in Figure Ten.

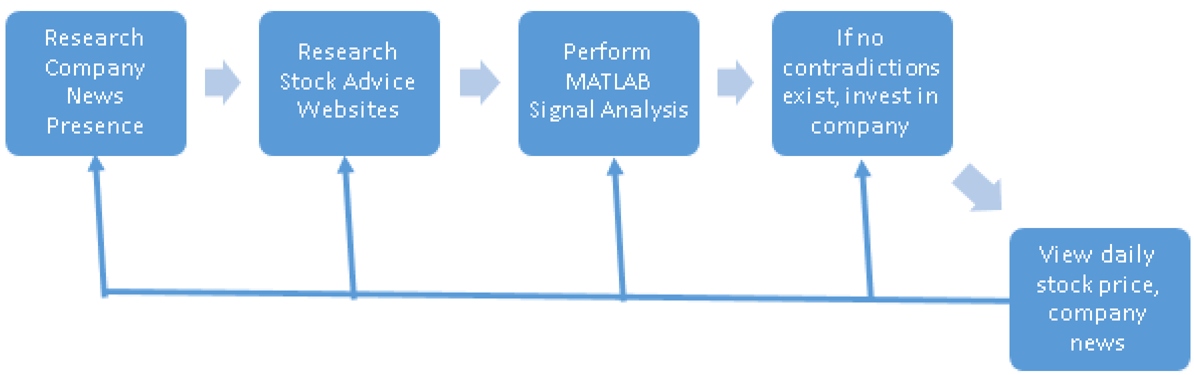

Figure 10: Three Stage System with Daily Feedback

Model Evaluation

Investment Simulation

Much of the testing for our models came through researching the stock market and virtually investing money. This was done several times with a theoretical $100,000 and with the simple goal of earning a profit. This is a very simple test to run: once analysis has been performed and stocks have been modeled and chosen for investment, all one has to do is keep up with the stocks and see how much money is made or lost over the course of the investing period. One of the most important things about this test is that it is very easy to determine exactly how well the models are working, since the result is a scalar value representing the offset from the original $100,000 investment. One either earns a profit or generates a loss, and the magnitude of this value determines how well that test went. This is also a downside to the test, as it only determines how well something performed, and does not tell why or how. The next important task of this test is to try to determine why one either made or lost money and that is an assessment for after the virtual investment.

Individual Aspect Testing

As part of the testing process to see the importance in all of the many factors that go into buying a stock, they were used almost independently to view their own potency as a model. This was done to determine a potential weight that should be put on the 3-part model. This was done by choosing several different sets of stocks. The first set was selected using only the confidence model. The next set of stocks were chosen by potency on the news along with being cross-referenced with the confidence model. The final set of stocks were chosen based on their rating on stock websites such as Forbes or InvestorPlace. This test is done to assist other unquantifiable methods in the decision making process.

Results

Term A

At the conclusion of Term A, the three basic analysis methods which provided the most accurate results were implemented in MATLAB using the function *WeekSixAnalysis*, which can be found in Appendix B. The function consumes one autocorrelation period of a stock and its exchange index, and produces six sets of 20 prediction values, which includes one month of prediction data for each of three models, as well as each model combined in weighted addition with the exchange index model. Also produced are a graph of the exchange index model, and graphs of each of the six model varieties superimposed on the actual stock prices, along with the percent error associated with each prediction. At this stage in the project, exchange index models were combined with the stock models by using volatility ratios, as described by Equation Six. Figure Eleven shows the linear regression and Fourier series model of the NYSE index along with daily actual index valuation at market close.

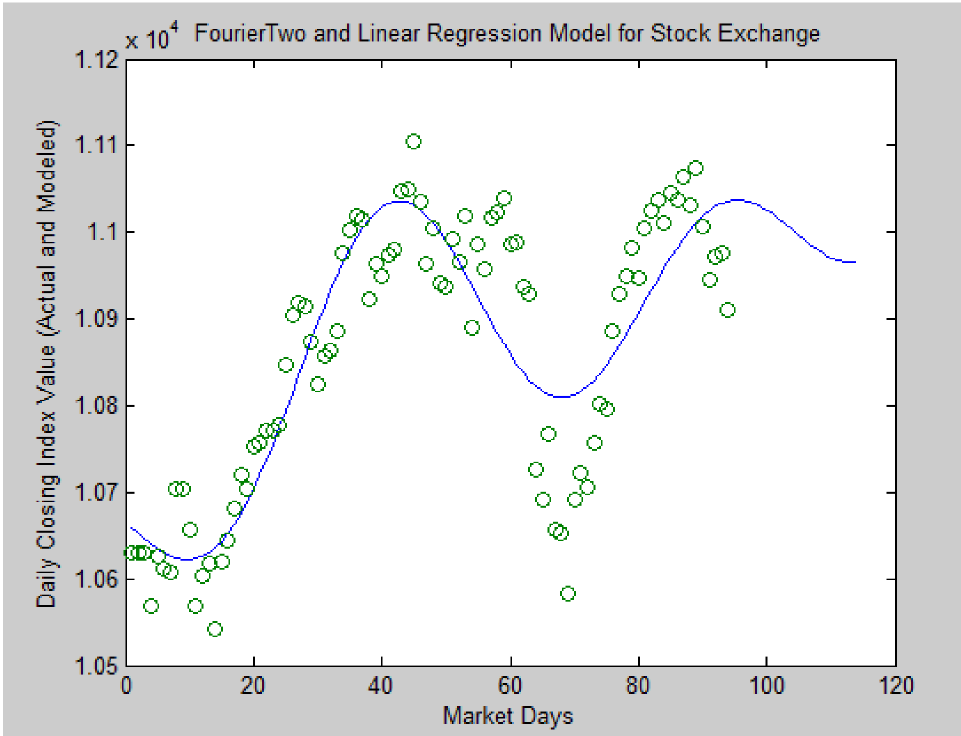

Figure 11: NYSE Index Model

Results of Term A analysis were somewhat mixed, with the six models providing varying levels of accuracy among the ten stocks. For Agilent Technologies (A, NYSE), FourierTwo and linear regression provided the best model out of the three base models with an error below 4%, and the addition of exchange data increased the error for all three.

The ARIMA(2,1,2) model proved to be most accurate for AMZN, a stock which had been relatively volatile during its autocorrelation period, but has become much more smooth in the past month. This resulted in the two models using Fourier analysis to be wildly inaccurate, predicting sinusoidal arcs where the signal had become mostly linear. Due to this high volatility, the inclusion of exchange index data resulted in more of the exchange model being used, which flattened out the sinusoids in these models, increasing their accuracy.

The most accurate model of INTC was the combined FourierTwo and ARIMA(2,1,2), which had an error under 5%. This error was slightly increased by the addition of market data. LPL, which had been on a bullish upward trend throughout its autocorrelation period, fell during the month of predictions. The Linreg-FourierTwo and ARIMA(2,1,2) models both incorrectly predicted the rise to continue, while the FourierTwo-ARIMA model accurately predicted the fall. The addition of exchange data increased the accuracy of the two incorrect models by reducing the upward slope of predictions, while decreasing the accuracy of the FourierTwo-ARIMA model.

For QCOM, the most accurate model was ARIMA(2,1,2) (less than 3% error), which was decreased slightly in accuracy by the influence of the exchange index. The most accurate model for STM was Linreg-FourierTwo, resulting in less than 8% error. This error was unchanged by the inclusion of exchange index data. For TQNT, the first two models were quite inaccurate, while the ARIMA-FourierTwo model predicted the future data points entirely within the noise bands. This accuracy was greatly reduced by the addition of NASDAQ index model data.

Texas Instruments is an example of a stock which was accurately modeled by all three analysis methods, with error under 5% for all three and under 2% for FourierTwo-ARIMA. This accuracy seemed unchanged by the addition of exchange index data. For VRTU, ARIMA(2,1,2) provided the most accurate results (under 6% error), with the other two models much less accurate. This level of inaccuracy was reduced by a factor of ½ when the exchange data was included in the models (8% to 4%). For WDC, none of the six models could accurately predict where the stock price was moving. This is likely the result of other outside factors affecting the performance of the company and/or its stock. Figure 12 shows the results of all six models for Agilent Technologies, as an example of the wide variation observed between six models for a given set of data.

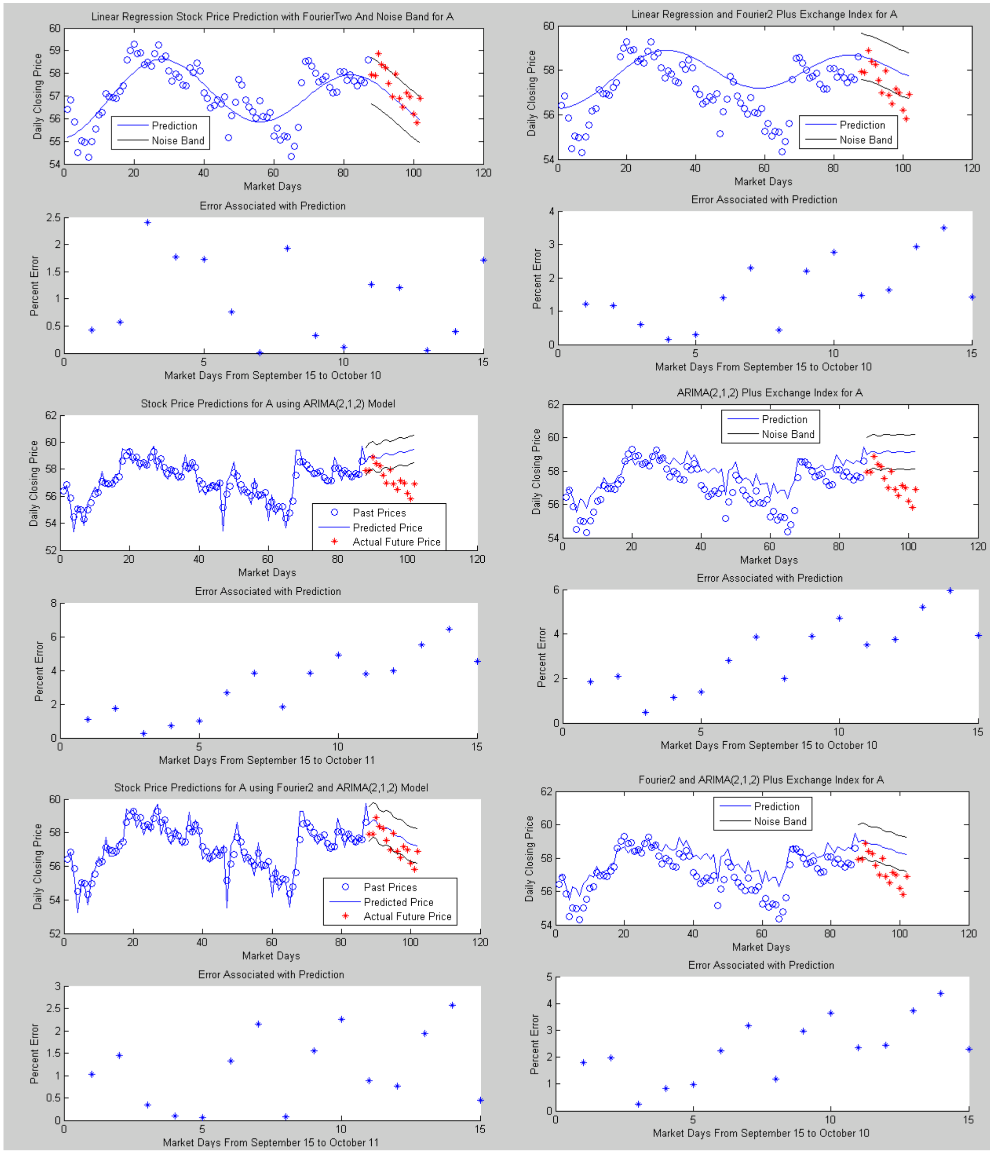

Figure 12: Six Term A Models, Agilent Technologies

In general for the Food Industry stocks, the basic Linreg-FourierTwo and the FourierTwo-Linreg performed the best and if one of them guessed wrong it was likely that the other one had a good prediction on what was happening for the stocks. These models seemed basic but with the noise band and a low number of Fourier coefficients used it turned out to be a good model. This is mainly true for PNRA, HSY, DNKN, DF, and SVU which in one of those models not only had 3% error but also followed the natural trend of the data really well and remained in the error bound really quite well. Stocks that generally didn't model well in this model where either of very low volatility or the model just couldn't predict the rise or drop properly.

Generally for the rest of the stocks that were actually modeled well was using the ARIMA method. These stocks were generally less volatile and didn't need any Fourier representation to model them. These stocks include: MCD, TSN, and WFM which all had ~3% error. Other than these stocks the data seemed to be too volatile to model using the ARIMA. This method also doesn't seem to model great drops and rises in the market which is how money would be made in this industry.

Other than those eight stocks that other two were not modeled very well since they seemed to be more spontaneous with what was going on as the future progressed. And most of the other models that were developed to try to solve this problem are not scaled properly so they are an improvement on the models that have 15% error but still do not follow the trend of the data properly, which is the main goal of these models.

Term B

December One-Month Portfolio Challenge

By the midpoint of Term B, the team had developed several models of stock signals which made use of historical stock prices and exchange index data, capable of producing short-term forecast values with varying levels of accuracy. In order to test these models, each member of the team performed independent analysis on stocks of their choice, and used the models to make investment decisions. A total of 100,000 USD was theoretically invested by each researcher, and each stock in the portfolio was chosen to be either bought on December 1 and sold on January 1, or to be short sold in that same period of time. In this manner, stocks whose models forecasted a rise in price could be bought and sold to turn a profit, while stocks whose models predicted loss could be short sold for the same potential gains.

Many stocks were analyzed in order to create portfolios that the team was confident would produce positive gains. Stocks which were forecasted by each of the models to increase in value over the relevant 24 market days were selected for purchase with a strong factor of confidence, and stocks which were forecasted by both models to decline in value were chosen to be short-sold with the same level of confidence. When the majority of models predicted a movement in one particular direction, positive or negative, the stock was chosen for investment, but with a lower level of confidence, with a smaller amount of capital invested in that stock. Stocks with results that varied greatly between the models were not chosen, due to the factor of uncertainty indicated by two conflicting models. Table One in Appendix A shows the stocks chosen for investment and profit/loss for each member of the team. Figure 13 shows the daily profit/loss values of Sidney's December $100,000 portfolio. This investment had a peak gain of 3.2% and a final profit of 1.1%, or $1,129.

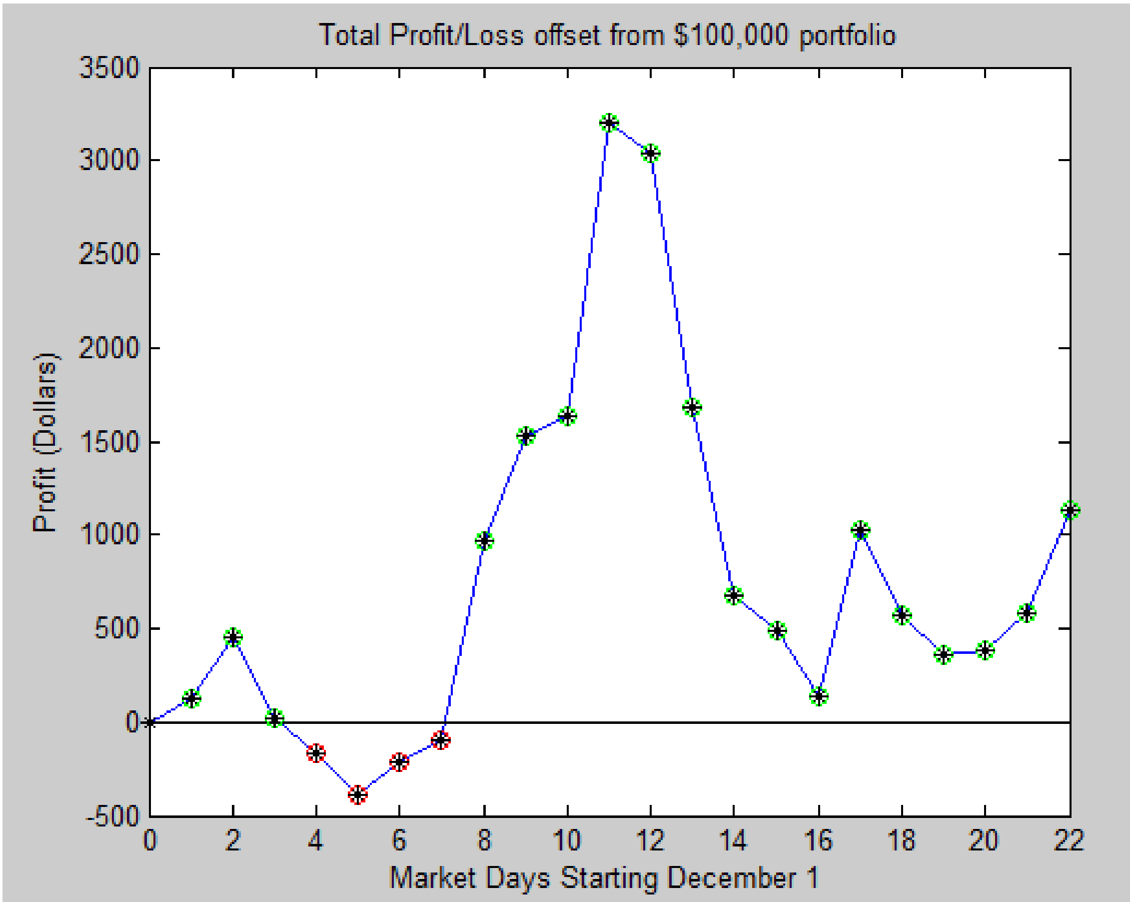

Figure 13: Total Portfolio Profit (Sidney, December)

Figure 14 shows the results of Jacob's December investment portfolio. The decision to invest in these stocks came from the confidence model using all six models that were developed. The trend of the stock was the most important aspect of the modeling process and the decision making process. The weight of each stock did not end up being calculated mathematically more so just found by making the total investment within the stock around $10000 to $20000. This was later found to be a flaw in the system but fixed for a future test. For the first half of the test the stocks waivered within a neutral zone not making or losing any significant amount of money as suspected by the models. But for the latter half of the month most of the stocks, as predicted, started to make money. This investment had a peak gain of 3.8% and a final realized profit of 2.3% or $2355. This turned out to be very good since all of the stocks were bought and made money during a difficult time for the market.

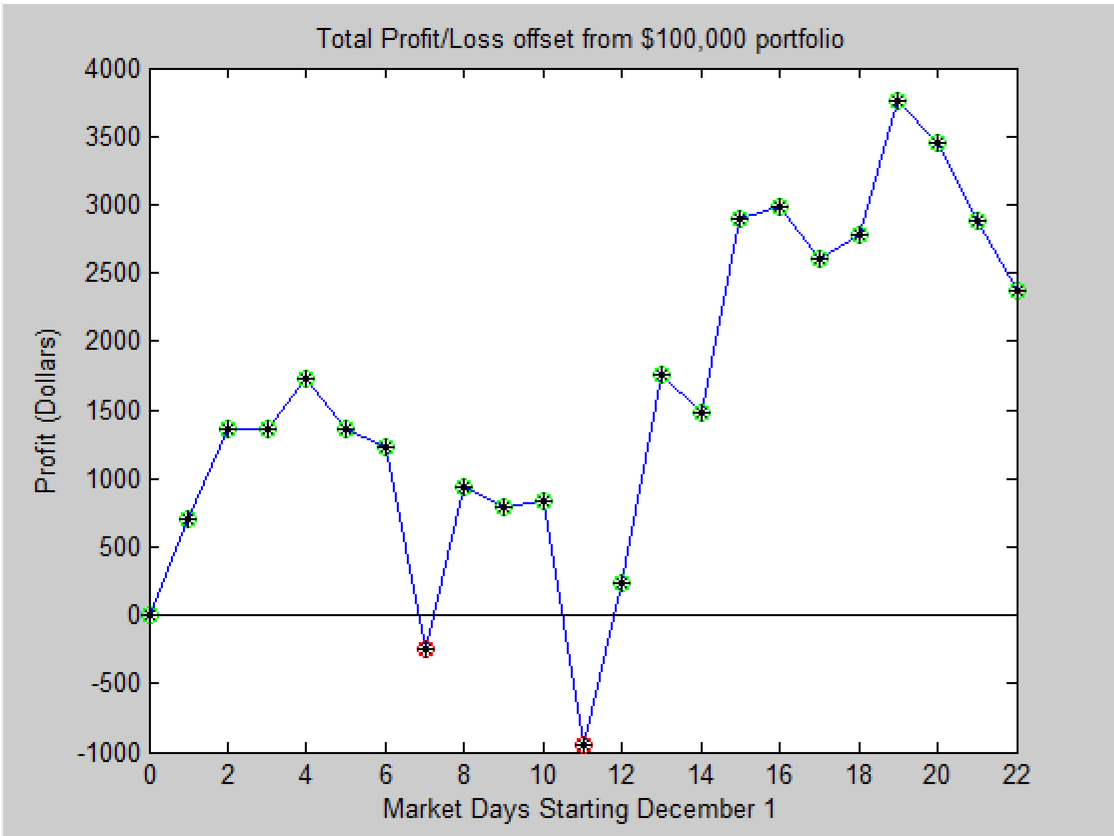

Figure 14: Total Portfolio Profit (Jacob, December)

Term C

January Portfolio

A second one-month investment challenge, in which the team was given the option of choosing a new set of stocks valued at $100,000 or continuing with the December stocks, began on January 1. Sidney's portfolio consisted of eleven new stocks ranging across different sectors of American industry, outlined in table three in Appendix A. Seven of the stocks were chosen to be bought, and four were chosen to be short sold. The predictions were made using the results of two models: the volatility-based model which chooses one of three analysis techniques for a given stock, and the exchange index model, which was combined with the volatility-based model using Equation Eight. This was implemented within MATLAB using the function *OneMonthChallenge*. Each stock was given an equal monetary weight in the portfolio. The results of this investment challenge, the contents of which remained unchanged for the duration of the month, can be seen in Figure Fifteen. Over the course of this investing period, peak gains were seen at $500, with peak losses much lower, at $1200. Total profit from this round of investing was $16.76, or %0.016.

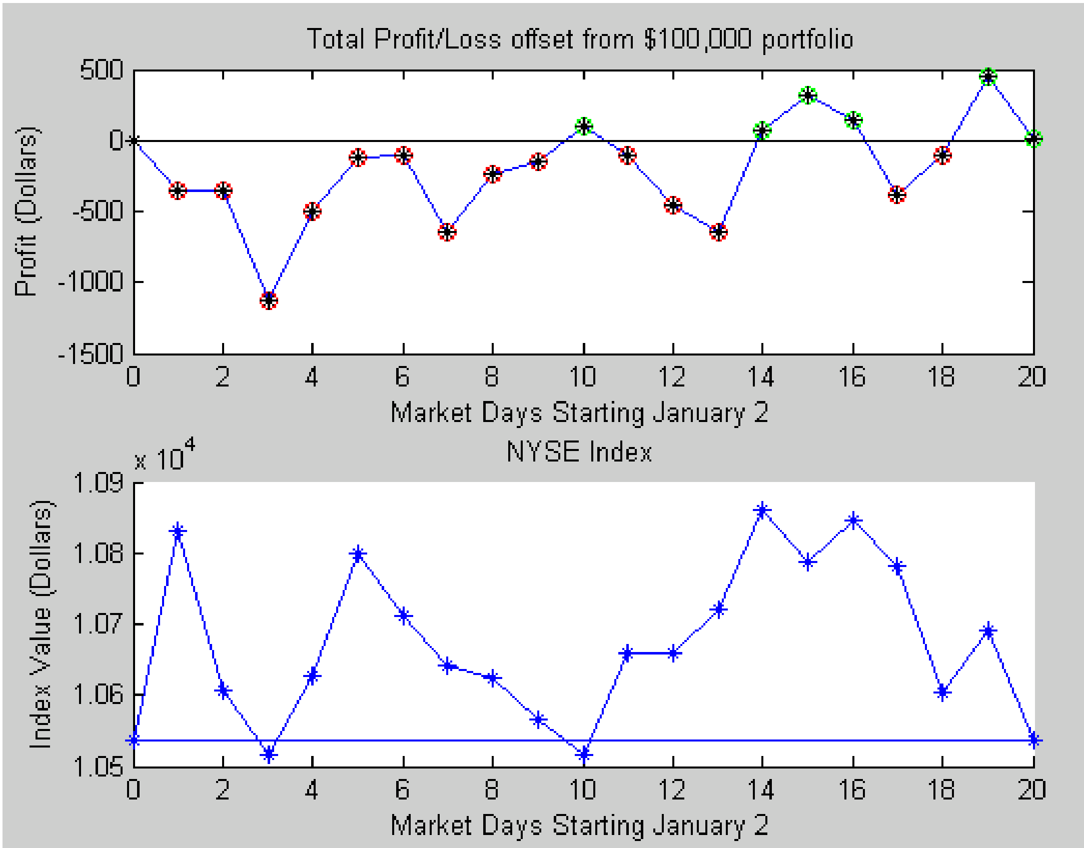

Figure 15: $100,000 Portfolio Offset (January, Sidney)

For the beginning of the December investment a 35 day model was applied to the stocks to make the decision on pick certain stocks. This meant that after the December investment was done the models were not fully tested. Since the stocks generally did very well throughout their initial 22 days, the decision was made to continue using the same stocks for the next month. Though the model did not last long enough to model all 41 days for the two months, but it was close enough to take the risk on the last six days. The results came out quite well with a peak gain of 6.8% and a total gain of 3.3% or $3301. As one may note, the peak price did take place as the model of the stock wore off. If the stocks were sold when the modeled ended more profit would have been made. Figure 16 displays the entire two month investment and the respective output profit for each given date. For the most part, the investments made money and acted as expected. They started out level and then eventually made a significant amount of money.

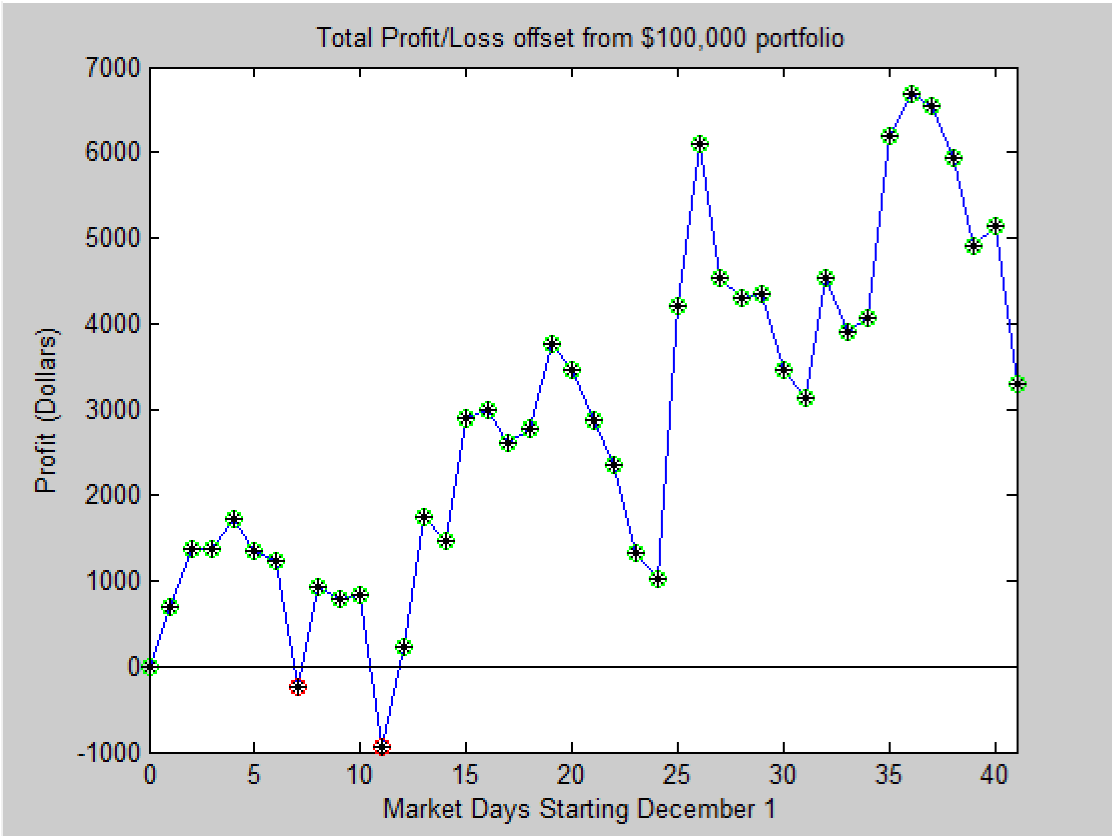

Figure 16: $100,000 Portfolio Offset (December & January, Jacob)

Research-Based Model Portfolio

Sidney Veilleux's Investments

At the midpoint of Term C, a new investment campaign was launched with the same goals as the previous two challenges: to utilize the models with the intention of earning a profit from stocks totaling $100,000 in value. This new portfolio was carefully constructed using the results of the research-based model developed during this term. Due to the more selective nature of this investment selection system, many more stocks were analyzed in order to find just a handful of viable investments. Of the thirty companies that were researched for Sidney's portfolio, only seven were able to pass all three stages of the selection process, which were subsequently chosen for investment. These stocks, which are listed in Appendix A, were given equal weight with the exception of one stock, which was given triple the monetary weight as the other six. The logic behind this decision was that of all the stocks researched, only one stock was recommended by the model to be short sold. In order to ensure that drastic changes in the stock market wouldn't skew portfolio results, more shares of this stock were short-sold. In this manner, the losses that would be incurred in the other six stocks by a large market crash would be more absorbed by gains in this stock (JPM, NYSE). The research-based model predicted stock prices for these seven securities over a period of one month. The results of this investment can be seen in Figure seventeen, which depicts profits and losses for Sidney's portfolio starting on January 26.

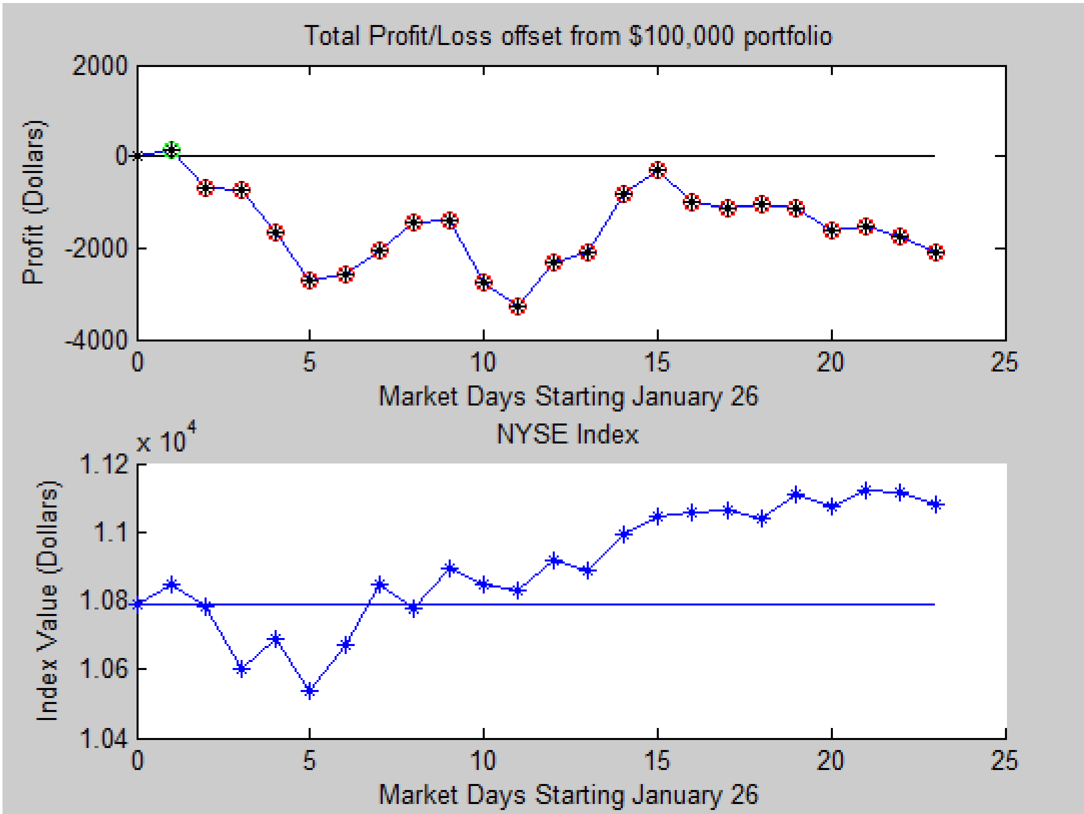

Figure 17: Research model results (Sidney)

As can be seen in the figure above, this portfolio generated a loss valued at $2073.37, or 2%. Table Five provides a detailed breakdown of the total profit/loss of the individual stocks in this portfolio. From the table it can easily be deduced that the majority of the losses in over this investment period came from the incorrectly-predicted rise in value of JPM. Four of the eight stocks generated moderate gains (average $654 per security, or 6.5%), while the remaining three (INTC, YHOO, LPL) produced losses averaging $700 per stock. Figure 17 also shows the value of the NYSE index over the duration of this investing period. Between days one and five, the stock market and portfolio fell in value at similar rates. Over the remainder of the month, the market recovered and surpassed its original value, while the portfolio failed to follow suit. This observed amount of increasing losses lead to team discussions about defining some method of determining when to sell off stocks which may only yield more losses in the future, and do the same for stocks which have far surpassed their intended level of gains. In this manner, portfolios could be reviewed daily as part of the feedback model, and investment decisions could be made based on the actual stock price compared to the forecasted price for that day.

Jacob Grotton's Investments

For the Term C investment challenge new stocks were selected, but this time with the goal of incorporating research into the decision making process. As a test, different techniques were used to select different stocks as shown in Table 1. In this manner, it could be determined which method of stock selection was most reliable over the same period of time, under the same market conditions.

Stock	Symbol	Decision Method
IPG Photonics	IPGP	Confidence Model
The Hershey Co.	HSY	Confidence Model
Sonic Drive-In	SONC	Confidence Model
Tesla Motors	TSLA	Stock News / Confidence Model
The Walt Disney Corp.	DIS	Stock News / Confidence Model
United Insurance Holdings Corp.	UIHC	Prophecy / Confidence Model
Biogen Idec Inc.	BIIB	Prophecy / Confidence Model

Table 1: Decisions for Research Portfolio (Jacob)

The first three stocks that were selected (IPGP, HSY, and SONC) were chosen purely due to the outcome of the confidence model developed in Term A. Since this was the only factor used in the selection of the stock, all six of the numerical models had to have the same trend for the stock in order for it to be selected for investment. This meant a much more exclusive selection process, as the level of confidence had to be 100% for a given stock. Next, Tesla Motors and Disney were selected based on their presence in the news and how their company was growing and developing. For Tesla, the news consisted of the company developing two new cars that have recently been released, and having the highest expected sales of any other car they have released before. Similar news was released about Disney, with many employees being added along with multiple new rides waiting to be added to many of their parks. These stocks were then tested using the confidence model to add confidence to the decision, but since they had research along with them, only four of the six models had to be following the same trend. The introduction of other decision-making tools allowed for a reduction of the amount of mathematical confidence required of the model, since more factors were at play. Lastly, the final two stocks (UIHC and BIIB) were selected based on suggestion of popular stock websites, including InvestorPlace and Forbes. This was to test the effectiveness of the "self-fulfilling prophecy" observed for many stocks which become successful due to their popularity, since many people view these sites and act on the advice of these analysts. Many of these stocks are added to the analysts' "buy" lists based on positive trends in their recent price history. The stock continues to rise as the analysts create a higher demand for it, and their predictions come true. The implementation of this concept as a means for investment decision-making meant that not as much confidence was required from the numerical models. Therefore, only four of the numerical models needed to match, as decisions were being made using models from professional stock investors as well.

In this investment challenge all of the stocks that were purchased made money and were successful. This was a very good sign for the addition of the two types of research model. For this portfolio of stocks, IPGP made the most money throughout the investment challenge. This stock ended up making $3567 which was due to a 31.6% increase in value of the stock. For the most part, all of the stocks did well and of the methods used performed roughly equally. This is a good sign, since most of the techniques are equally balanced. With the peak profit of 8% and the final profit of 7.6% or $7580. This was a very successful simulation since most of the past attempts averaged 3% gains.

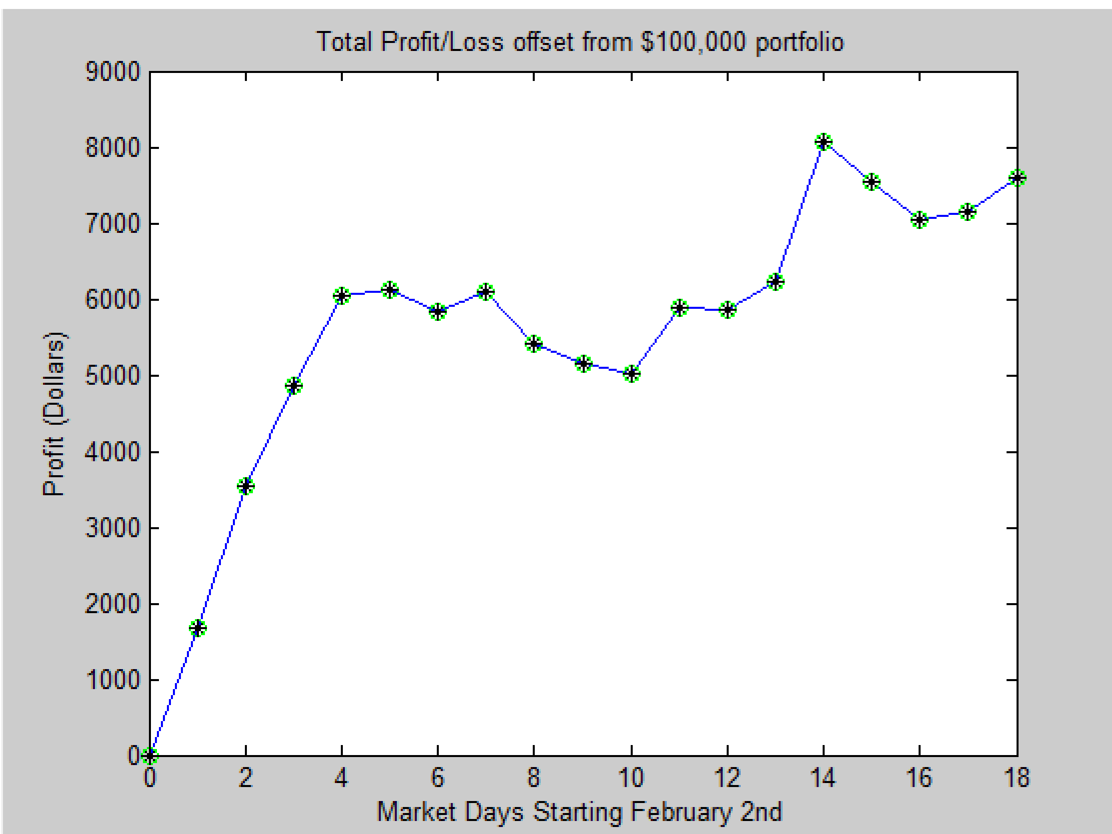

Figure 18: Research model results (February, Jacob)

Nanotechnology Stocks

Application

At the midpoint of Term C, the team had produced a fully-developed system capable of providing investors with informed financial decisions from a variety of input factors. This system had been put to test on high-value portfolios containing wide ranges of stocks, but were yet to be tested at the smaller scale of investments in a single stock. Many companies in the field of nanotechnology were chosen for research and used as inputs to the modeling system. This field is a rapidly expanding sector of American industry, divided into three major parts: nanomedical devices, nanomaterials, and semiconductor research. Two of these fields were chosen by members of the project team, and a large list of companies was narrowed down to several stocks which passed all the tests of the model.

Graphene

Graphene is a relatively new material that is widely considered to be one of the best electrical conductors ever developed. It consists of layers of graphite whose molecules are arranged in such a way which yields very high strength and very high conductivity. This material is expected to revolutionize the field of electronics, due to this durable and conductivity, as well as its very low molecular density. This is a very good sign for things to come as far as the stocks of companies that produce or use graphene are concerned. Currently, graphene is not practical for use in technology due to the fact that it is currently very difficult to produce. The cost of graphene production far outweighs the money saved by using it as a replacement for other materials.

This current situation makes graphene stocks a perfect opportunity to invest in now. Stocks of companies that either use or produce graphene and carbon products will increase greatly in the future, when this material becomes cheaper to produce. This implies that these stocks are currently undervalued, which makes them excellent candidates for long-term investment, as they have nowhere to go but up. Most carbon manufacturing companies are expected to perform very well once graphene is easy to produce, as the demand for this useful nanomaterial skyrockets. Companies that currently use this technology will also increase in value, as it will be cheaper for them to manufacture products containing graphene. One of the main companies that is expected to use more graphene once it is cheaper is Tesla Motors. They are in the automobile industry, and are already using graphene as a protective outer shell for their vehicles, as well is for in-dash touch screens. This company currently invests a lot of money into graphene research and takes pride in having the newest technology in their cars.

One thing to note about graphene is that it is certainly a long term investment so until it is easy and cheap to produce the stocks will not be affected at all. Once it is easy to produce, all the stocks from electronics companies will start performing really well, since this superior conductor will replace much of the metal in chips, reducing their power consumption. It is predicted that all stocks will benefit from a cheap source of this super material.

Hexcel was the other graphene stock that was watched very closely. It is a company that produces carbon-based materials for the production of wind turbines and other green energy applications. The price of this security has been rather low in the past few years, but it is currently on the rise, increasing 25% over the last 50 market days. This company would certainly benefit from the development of inexpensive graphene production, since it will be able to respond to increased demand in a short period of time. Currently, Hexcel is doing very well due to the ease of production of carbon nanotubes. This company's success can be seen in the graph of its stock price, shown in Figure 19.

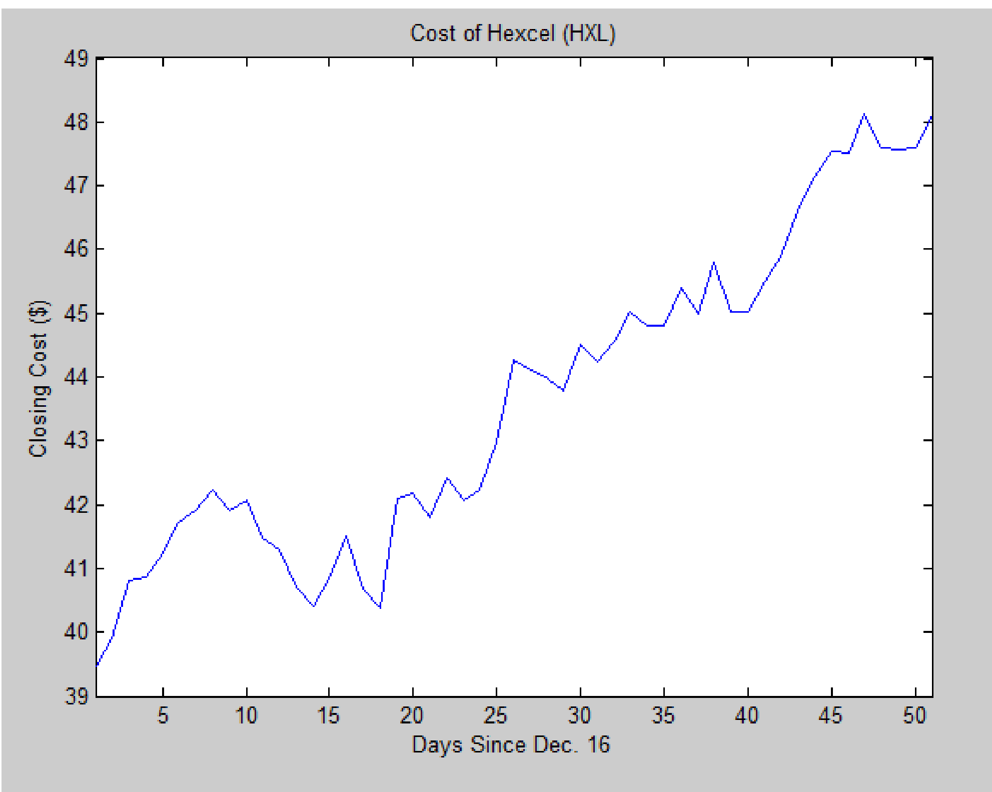

Figure 19: Price of Hexcel (HXL, NYSE) Starting Dec. 16th

As one can see in the figure, this stock has been doing very well recently. This is likely due in part to the recent expansion of the stock market in aggregate, which has effected most stocks over the past two months. This is also the most recent amplitude spike since the last, which occurred in 2008 when carbon nanotubes first became popular. This is a very good sign for things to come, since Hexcel responds very rapidly to advancements in its field of technology.

Semiconductor Technology

A field that is on the forefront of research and development in the nanotechnology sector is semiconductor technology. Many companies exist with the goal of developing methods for producing smaller semiconductor process nodes for integrated circuits (ICs) and using these methods to manufacture ICs on a large scale. Moore's Law states that due to this research, the amount of silicon transistors that occupy a single area on a chip doubles every two years. Accordingly, companies in this industry are constantly working to beat their rivals, always working to produce ever smaller nanotechnology circuits.

One company that passed the three steps of the research-based modeling system is Taiwan Semiconductor Manufacturing Corp (TSM, NYSE). A news search for the company revealed nothing but positive results: they had recently purchased $17M worth of new manufacturing equipment, and had also recently announced very strong Q4 2014 earnings due to increased iPhone sales (they have contracts to manufacture integrated circuits for Apple). The opinions of market analysts at TheStreet and MarketWatch were polled, and both gave TSM a "buy" rating. The historical stock price data was fed into the MATLAB model, and numerical analysis predicted a very high increase in price over the coming month (nearly 40%). This stock was added to the research-based portfolio as a sign of good faith in the predictions of the model. Over the course of the investing period, the stock value remained below the predicted values, only slightly higher than the original started price. The investment results for this stock can be seen in Figure 20. Total profit generated from TSM investments was valued at $287.73, which corresponded to a 2.8% increase.

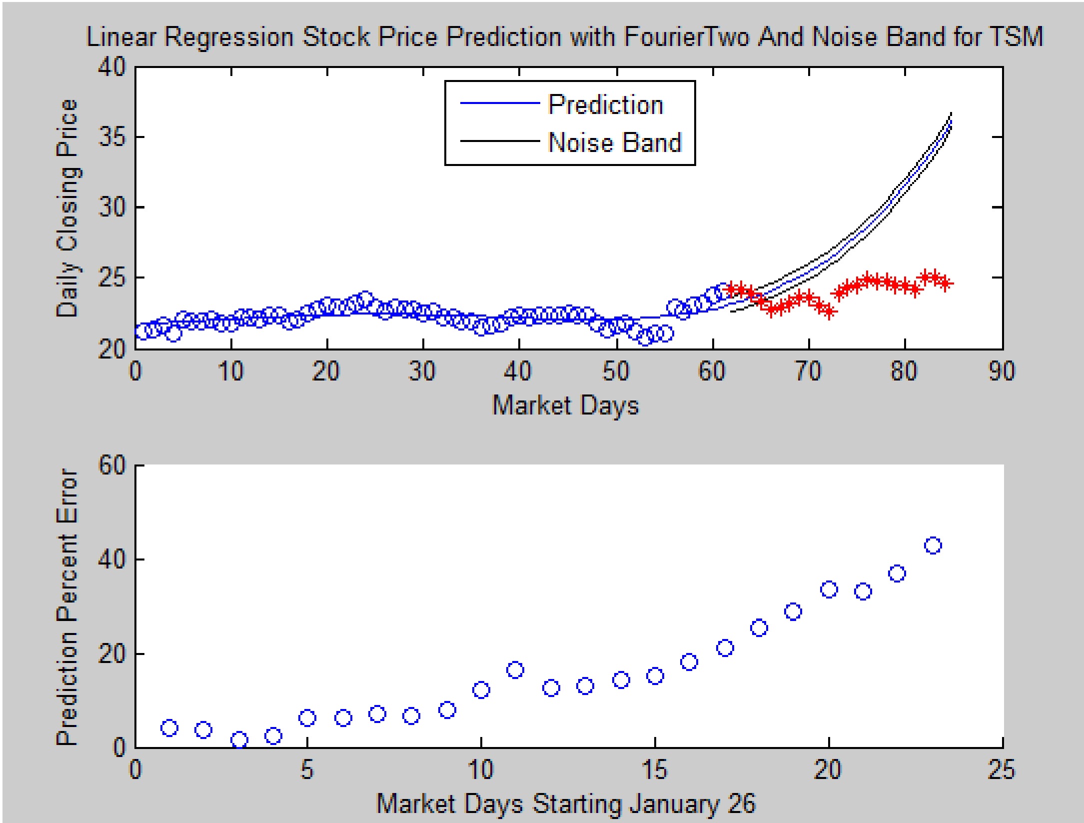

Figure 20: TSM Results

Another semiconductor company that is expected to be a profitable investment is Applied Materials (AMAT, NASDAQ). AMAT is a major producer of semiconductor manufacturing equipment, supplying major IC fabricators with the latest technology to produce smaller process nodes. In the news, this company has recently announced a merger with their fiercest competitor, Tokyo Electron. The elimination of competition means that they will no longer have to undersell their products, increasing profits. Market analyst websites TheStreet and MarketWatch were consulted, both recommending a "buy" of this security. The stock data was fed into the MATLAB model, and an increase was predicted over the next 20 market days. For these reasons, this stock is recommended by the team for short-term investment. The numerical model predictions and actual results for the first eight market days can be seen in Figure 21, along with the associated error of predictions. So far, this stock has outperformed the predicted values, though its price still remains within the noise band.

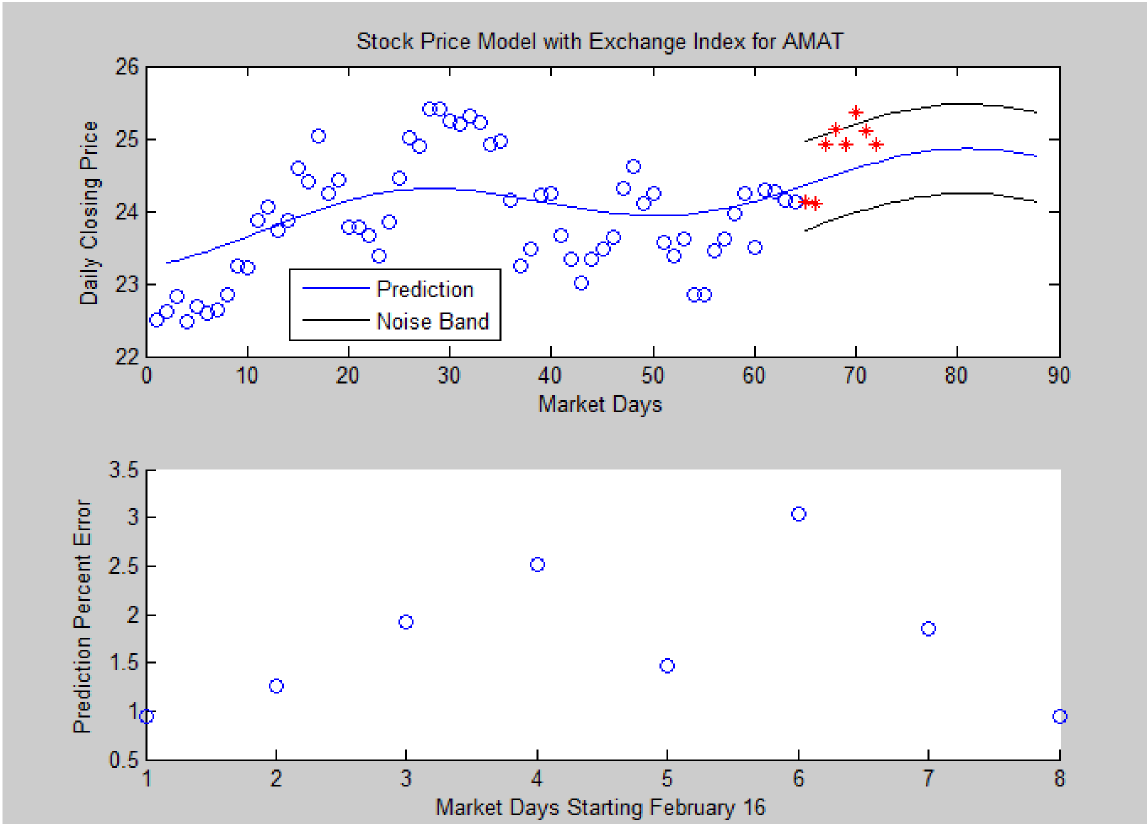

Figure 21: AMAT Model and Results

Discussion

Term A

Overall, the FourierTwo and ARIMA(2,1,2) model proved to be most accurate base models used in Term A, having the lowest percent error for 50% of the ten chosen stocks. This model uses Fourier Analysis to remove the sinusoidal component of the signal before performing ARIMA analysis, then adds the two analyses methods together to create the model. In cases where this model accurately predicted future prices, the inclusion of exchange index data only seemed to decrease this accuracy. However, in cases where any of the three models provided inaccurate predictions, the addition of the exchange index model reduced this inaccuracy, resulting in better models.

All three models provided accurate results for some stocks while producing inaccurate predictions for others. As the first term of the IQP came to a close, the team had produced three relatively accurate models for predicting the price of a stock, each of which lack the consistency of providing good predictions across all ten stocks in the portfolio.

It should also be noted that on October 10, 2014, the New York Stock Exchange fell 1.12% and the NASDAQ fell 2.33%. This is indicative of all ten stocks falling drastically in price on that day. This caused nearly every model of every stock to be incorrect in predicting the price of the stock for this final day of analysis, since outside market factors affected the price of each of the ten stocks. These types of macroeconomic events, which are beyond the scope of this time signal analysis IQP, can be impossible to predict, especially weeks in advance. There are many outside factors that can affect the movement entire markets of stocks in aggregate, both positively and negatively, that can render even the best time signal model useless. One way to avoid loss during these drastic market shocks is diversification. Investing in multiple market sectors and even multiple markets, including global securities overseas, can help reduce loss when one market sector suffers unpredictable loss. This problem was revisited in Term C with the introduction of the research-based model.

Term B

Over the course of Term B, many new equations were developed in the search for a reliable method of combining the stock price and exchange index models. These equations were all designed following the assumption that there existed some universal constant or constants that could be applied to all stocks equally, yielding accurate results for any security in any market sector. The analysis performed using these equations has proven that these constants may not be as universal as was originally assumed, as the results were generally rather mixed. The same iterative analysis was performed with each equation to determine the best scalar values for a set of ten stocks. These calculated values were then tested on another set of stocks in a different market sector, to verify the accuracy of the new models.

Many implementations produced positive change in regards to forecast accuracy for some stocks, while reducing the accuracy for others. Some implementations provided excellent results for the ten stocks during iterative analysis, while yielding little actual change to the base model when applied to the ten test stocks, due to the very small-magnitude alpha values which usually resulted from experimentation. Due to the overall mixed nature of results from each model implementation, it was concluded that one of the following two statements must be true: either none of the models are accurate for predicting the price of a stock, or the values of alpha and beta are not as universally constant as was previously assumed. For future analysis, it was recommended that new equations be developed, in which constants alpha and beta are determined based on other stock metrics, including perhaps the volatility and volume values for individual stocks.

Over the first few weeks of the December One Month Portfolio Challenge, most of the earned gains in Sidney's portfolio were caused by forces beyond the scope of the project, as US markets fell by nearly 4% over the course of one week. This resulted in the loss in value of many stocks, including those that Sidney had chosen to short sell for the month of December. This market retraction was also the cause of the majority of portfolio losses this week, as stocks which he had chosen to buy fell by several percentage points. The events of this week can be viewed as an excellent argument for the equal distribution of securities to be purchased and securities to be short sold in a portfolio. In this manner, if the market moves unpredictably and drastically in one direction or another, large gains can offset relatively equal losses. This is the basis for the reasoning behind hedging, a strategy commonly employed by traders to ensure that unpredictably large losses are offset by *relatively equal* gains. Of course, the focus of this IQP is to develop a stock model for forecasting future prices. However, the past four months of analysis have shown that no model is without error, and intelligent trading strategies must be employed if the intent of the trader is to turn a profit.

Term C

The main goal of team efforts for Term C was the implementation and testing of a research-based model, which took into account the numerical methods developed in previous terms, along with the interpretation of real-world company news. Once developed, the models were tested using a $100,000 portfolio investment challenge. Many of the stocks in these portfolios were forecasted accurately (11 out of 15), resulting in capital gains. The four stocks which generated losses were all a part of Sidney's portfolio. These stocks generated consistent losses throughout the investing period, which lead the team to discuss methods for determining a proper course of action for stocks which were predicted entirely incorrectly by the model. This method of decision-making should also take into account the opposite case: what to do with a stock which is outperforming its prediction values? Two solutions were developed, though neither was tested on an investment portfolio due to the lack of time for a new investing period at the conclusion of Term C.

The first system developed for making premature buy/sell decisions during an investing period is the feedback model, which is discussed at length in the Methods section of this report. This model allows the investor to check the daily news on a company, as well as the changing opinions of market analysts, in order to make informed decisions on whether or not to change the quantity of shares currently held in a stock. This method would allow for quick reactions to bad news about a company, saving the investor from potential losses over the remainder of the original investing period. This would have saved Sidney's portfolio from a large portion of its losses, as JPM announced high earnings and increased profits shortly after the portfolio challenge had begun. However, this system is not without flaw, as it does not take into account the commission cost of buying and selling stocks on a daily basis, instead of making only two transactions for each stock, at the beginning and end of the investing period.

The other method for making mid-investment changes to a portfolio takes into account only the difference between predicted and actual stock prices on a given day. This system makes use of a numerical threshold for deciding whether to liquidate an investment entirely for one of two reasons: either it has earned much more than the model predicted and the investor should cash in, or it has lost too much money and should be liquidated to eliminate further losses. This threshold is defined by Equation 12, where *buy_or_sell* represents a value of 1 if the stock was bought and -1 if the stock was short-sold. If the value of x is greater than 5%, the stock is recommended for liquidation due to the realization of gains much higher than expected. If x is less than -5%, the stock is recommended for liquidation due to the generation of losses greater than this acceptable level. This method is preferable to the feedback model, as it does not apply the use of daily trades, which wastes a considerable amount of capital on nonrefundable commissions.

$$x = \frac{price - prediction}{price} * 100 * buy\_or\_sell$$

Equation 12: Threshold for Liquidation of Individual Investments

Overall, the research-based model proved to be a more reliable form of investment decision-making, which can be seen clearly by the increase in profits for Jacob's portfolio of this investing period compared to the portfolios of Term B. The fully-developed three-part model is not without error, but it has been observed to generate more profit than loss overall. For future research, it is recommended that this model be tested further with more one-month investing periods to determine if this trend of profit is truly a result of the team's research, and not just good fortune.

The use of this threshold system would have greatly altered the outcome of Sidney's research-based investment portfolio. In the final week of the investing period, this threshold-based advice system was applied to the portfolio, and the model recommended selling several of the stocks due to high levels of loss. One stock which would have been sold immediately due to the recommendation of this system is INTC. This stock (which was a recommended buy) fell by nearly 10% within the first two days of investing. The system would have recommended a sell after day two, preventing further losses from being realized. The full month of results for INTC can be seen in Figure 22.

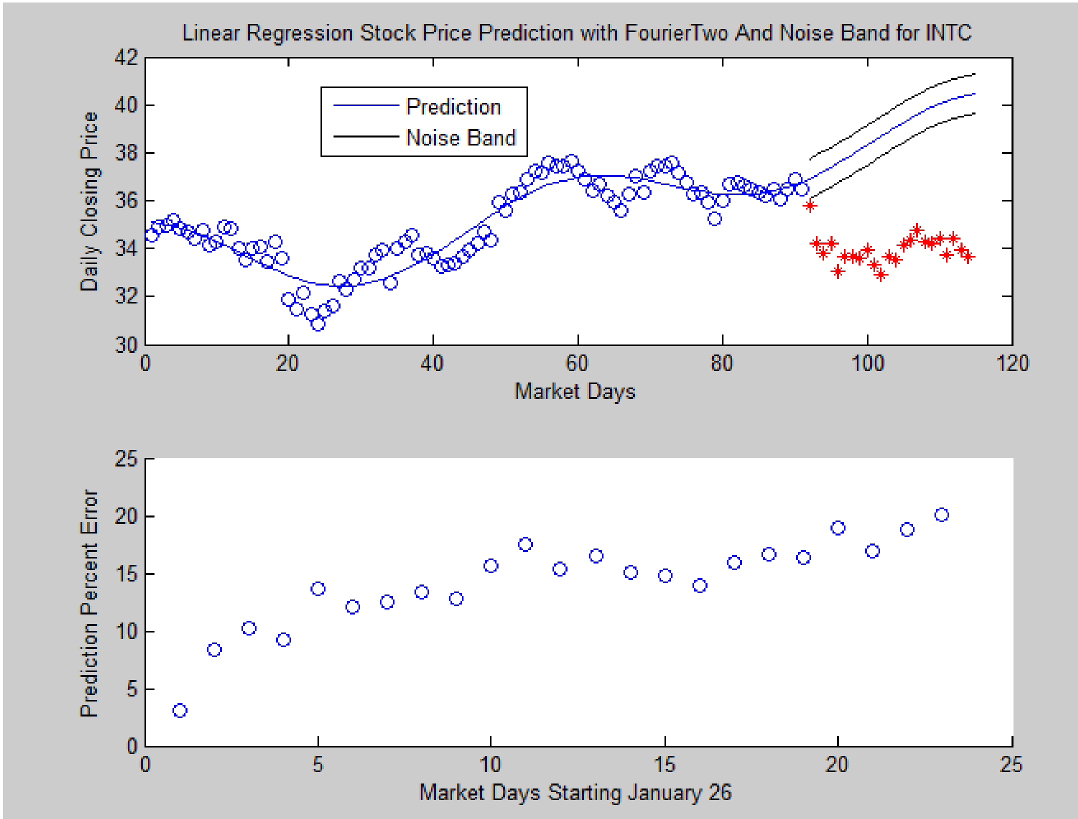

Figure 22: INTC Results

Conclusion

Over the course of this project, the team implemented knowledge of signal analysis techniques and the stock market to implement a fully-functioning system for providing investors with financial advice. This system made informed decisions using input from historical stock price data, historical stock exchange index values, trading volume, research into company news presence, and the opinion of market analysts. Several rounds of testing with virtual investment portfolios yielded mixed results, with a general trend of the system producing profitable outcomes for short-term investments. Though the price of a stock is often considered to be a seemingly random signal, with countless factors determining its future outcomes, the team has developed a system which is able to provide positive outcomes for even the most inexperienced of investors.

This IQP served as an incredible learning experience for the team, a group which had very little knowledge of the stock market at the beginning of the project. Over three terms, research was performed into many facets of the American stock market, providing insight about stock trading which will prove to be very useful in the years to come. Not only did the project yield a system which has proven to be profitable for short term investments, but it has also left the team with a wealth of knowledge about different metrics for describing stocks. This acquired stock market intelligence will greatly help in making long-term retirement investment decisions. The project has also strengthened the team's understanding of discrete signal analysis. Though both researchers have enrolled in signal processing courses, the application of these concepts in the world of finance has allowed us to better visualize the topics presented in class, putting real value to these mathematical methods of analysis.

For future teams embarking on this same path, it is recommended that the research of this project be taken into consideration and used as a starting point. The models developed in Terms A and B were the results of much testing and experimentation using a wide range of models, during which many less-accurate analysis techniques were eliminated. Future projects should focus on the creation of a system to quantify the research component of the system, with the intent of automating the entire process. If company news stories were to be downloaded, rated for positive or negative impacts on the stock price, and summed, the entire system could be executed simply by entering a stock symbol into a MATLAB script and receiving an investment decision moments later. This would provide the intended end-user (uninformed new investors) with an easy-to-use system for receiving financial advice.

In addition, the feedback and threshold models should be tested to determine which of these systems produces the most capital gains and the least amount of loss. These method were developed near the end of Term C, and the timing constraints of the project cut short the potential for another investing period. In the tests of these two systems for trading part-way through an investing period, there should be an estimated commission fee for each transactions, so as the better model the real-world issue associated with making frequent trades: it can be much more expensive than simply buying a stock and selling it one month later, especially if trades are being made daily. It may be discovered that implementing these systems would cost more in commission fees than the extra profit they generate, and thus should not be implemented at all.

Citations

"Graphene – What is it?." *A detailed decription of Graphene*. De La Fuente, Jesus., n.d. Web. 26 Feb. 2015. **http:// www.graphenea.com/pages/graphene#.VPI9RuHGoWs**

"How To Identify Patterns in Time Series Data: Time Series Analysis." *How To Identify Patterns in Time Series Data: Time Series Analysis*. N.p., n.d. Web. 11 Oct. 2014.**http://www.statsoft.com/textbook/time-series-analysis**

"Introduction to ARIMA Models." *Introduction to ARIMA Models*. N.p., n.d. Web. 11 Oct. 2014.

http://people.duke.edu/~rnau/411arim.htm

"MarketWatch" *Stock Quotes, Market Analysis*. N.p., n.d. Web. 20 Feb. 2015.

http://www.marketwatch.com

"Moore's Law." *Moore's Law Web page*. N.p., n.d. Web. 23 Feb. 2015. **http://www.mooreslaw.org/**

"Random Walk Model." *Random Walk Model*. N.p., n.d. Web. 11 Oct.

2014.**http://people.duke.edu/~rnau/411rand.htm**

"Stock Market Index." *Wikipedia*. Wikimedia Foundation, 10 Nov. 2014. Web. 11 Oct. 2014.

http://en.wikipedia.org/wiki/Stock_market_index

"TheStreet." *Business News that Moves Markets*, 20 Feb. 2015. Web.

http://www.thestreet.com

"Volatility Definition | Investopedia." *Investopedia*. N.p., n.d. Web. 11 Oct. 2014.

http://www.investopedia.com/terms/v/volatility.asp

Appendix A: One Month Challenge Results

December Challenge

Table 2: Sidney Veilleux - December

Sidney - December							
Company Name	**Symbol**	**Price on November 28**	**Predicted Price**	**Conclusion**	**Quantity**	**Total Investment**	**Profit**
Agilent Tech	A	42.74	25.33	Strong Sell	492	12462.36	$885.60
Ebay	EBAY	54.88	82.1	Strong Buy	226	12402.88	$280.24
LG Display	LPL	15.32	13.31	Sell	469	6242.39	$79.73
Sandisk	SNDK	103.46	89.53	Sell	69	6177.57	$378.12
STMicroelectronics	STM	7.48	6.67	Strong Sell	1874	12499.58	$18.74
Triquint Semiconductor	TQNT	24.37	18.12	Sell	350	6342	-$1,113.00
Virtusa	VRTU	40.07	47.6	Buy	156	6250.92	$249.60
Exxon Mobil	XOM	90.54	95.25	Strong Buy	138	12494.52	$263.58
Spectra Energy	SE	37.88	26.1	Sell	239	6237.9	$377.62
Dynegy Corp	DYN	33.15	30.58	Sell	204	6238.32	$571.20
Coca Cola	COKE	94.56	102.21	Strong Buy	132	12481.92	-$861.96
Total						**99830.36**	**1129.47**

Table 3: Jacob Grotton - December

December							
Name	**Stock**	**Cost**	**Shares**	**Investment**	**Cost**	**Investments**	**Stock Profit**
IPG Photonics	IPGP	$72.09	150	$10,813.50	$74.92	$11,238.00	$424.50
The Hershey Company	HSY	$100.28	75	$7,521.00	$103.93	$7,794.75	$273.75
ConAgra Foods	CAG	$36.52	400	$14,608.00	$36.28	$14,512.00	$96.00
Sonic Drive-In	SONC	$27.19	500	$13,595.00	$27.23	$13,615.00	$20.00
Ross Stores	ROST	$91.48	200	$18,296.00	$94.26	$18,852.00	$556.00
Lowe's	LOW	$63.83	150	$9,574.50	$68.80	$10,320.00	$745.50
Cato Corporation	CATO	$40.12	100	$4,012.00	$42.18	$4,218.00	$206.00
Dollar Tree	DLTR	$68.36	200	$13,672.00	$70.38	$14,076.00	$404.00
Medtronic	MDT	$73.87	107	$7,904.09	$72.20	$7,725.40	$178.69
			Total	$99,996.09	**Current**	$102,351.15	
			Remaining	$3.91	**Profit**	$2,355.06	2.36%

January Challenge

Table 4: Sidney Veilleux - January

Sidney - January							
Company Name	**Symbol**	**Price on December 31**	**Predicted Price**	**Conclusion**	**Quantity**	**Total Investment**	**Profit**
Briggs and Stratton	BGG	20.42	47.87	Buy	190	9095.3	-463.6
Disney	DIS	94.16	92.17	Sell	99	9124.83	370.26
Dunkin Donuts	DNKN	42.65	44.45	Buy	205	9112.25	805.65
Ellington Financial	EFC	19.96	31.94	Buy	285	9102.9	247.95
Goldman Sachs	GS	193.77	196.3	Buy	46	9029.8	-1071.8
NVIDIA Corp	NVDA	20.04	22.54	Buy	403	9083.62	-467.48
New York Times	NYT	13.22	14.09	Buy	645	9088.05	-503.1
Ratheon Co	RTN	108.29	114.62	Buy	79	9054.98	-776.57
Tesla Motors	TSLA	222.4	196.15	Sell	46	9022.9	852.38
Whole Foods Market	WFM	50.42	44.92	Sell	202	9073.84	-343.4
Yahoo	YHOO	50.51	48	Sell	189	9072	1366.5
						99860.47	16.79

Table 5: Jacob Grotton - December & January

January								
Name	**Stock**	**Cost**	**Shares**	**Investment**	**Cost**	**Investments**	**Stock Profit**	
IPG Photonics	IPGP	$72.09	150	$10,813.50	$74.84	$11,226.00	$412.50	
The Hershey Company	HSY	$100.28	75	$7,521.00	$103.21	$7,740.75	$219.75	
ConAgra Foods	CAG	$36.52	400	$14,608.00	$36.12	$14,448.00	$160.00	
Sonic Drive-In	SONC	$27.19	500	$13,595.00	$30.27	$15,135.00	$1,540.00	
Ross Stores	ROST	$91.48	200	$18,296.00	$92.11	$18,422.00	$126.00	
Lowe's	LOW	$63.83	150	$9,574.50	$67.76	$10,164.00	$589.50	
Cato Corporation	CATO	$40.12	100	$4,012.00	$42.55	$4,255.00	$243.00	
Dollar Tree	DLTR	$68.36	200	$13,672.00	$71.31	$14,262.00	$590.00	
Medtronic	MDT	$73.87	107	$7,904.09	$71.45	$7,645.15	$258.94	
			Total	$99,996.09	**Current**	$103,297.90		
			Remaining	$3.91	**Profit**	$3,301.81	3.30%	

Research-Based Challenge

Table 6: Research Challenge Results (Sidney)

Company Name	Symbol	Price on January 23	Predicted Price	Conclusion	Quantity	Total Investment	Total Profit
Intel	INTC	36.45	40.39	BUY	274	9,987.30	-767.20
Yahoo	YHOO	48.95	59.65	BUY	204	9,985.80	-918.00
New York Times Inc	NYT	13.11	13.22	BUY	763	10,002.93	816.41
LG Display	LPL	16.20	16.31	BUY	617	9,995.40	-413.39
JP Morgan	JPM	56.68	51.98	SELL	529	29,983.72	-2,592.10
Nvidia	NVDA	20.71	38.43	BUY	483	10,002.93	719.67
Texas Instruments	TXN	55.06	88.23	BUY	182	10,020.92	793.52
TSMC	TSM	23.99	35.23	BUY	417	10,003.83	287.73
						99982.83	-2073.36

Table 7: Research Challenge Results (Jacob)

Name	Stock	Cost	Shares	Investment	Cost	Investments	Stock Profit
IPG Photonics	IPGP	$74.64	150	$11,196.00	$98.42	$14,763.00	$3,567.00
The Hershey Company	HSY	$102.23	200	$20,446.00	$103.63	$20,726.00	$280.00
Sonic Drive-In	SONC	$30.27	150	$4,540.50	$32.60	$4,890.00	$349.50
Tesla Motors	TSLA	$203.60	100	$20,360.00	$207.19	$20,719.00	$359.00
The Walt Disney Comp.	DIS	$90.96	103	$9,368.88	$104.56	$10,769.68	$1,400.80
United Insurance Holdings Corp.	UIHC	$24.43	200	$4,886.00	$24.70	$4,940.00	$54.00
Biogen Idec Inc.	BIIB	$389.16	75	$29,187.00	$410.10	$30,757.50	$1,570.50
			Total	$99,984.38	**Current**	$107,565.18	$7,580.80
			Remaining	$15.62	**Profit**	$7,580.80	7.58%

Appendix B: MATLAB Code

Term A

Sidney Veilleux

```
%Sidney Veilleux
%Week Six Model for Stock Price Prediction
%This function consumes a real-valued periodic signal one period in length
%A String representing Stock Symbol
%A vector representing four weeks of actual values for determining prediction
accuracy
%A vector representing one period of Stock Exchange index values
%Performs calculations to predict four weeks of future prices:
        %Creates Fourier2 + Linear Regression Model for Stock and For
Exchange Index
        %Creates ARIMA(2,1,2) Model
        %Creates Fourier2 + ARIMA(2,1,2) model
        %Calculates Volatility of Stock and Exchange Index
        %Integrates exchange model into stock price model using volatilities

%Plots results of six model types along with prediction percent error




function y= weekSixAnalysis(x,r,s,exchange)
hold off;
N=length(x);   %This is the autocorrelation value
%Populate some vectors which are iterated over in a for loop
noise=zeros(1,N);    %noise vector for each fourier series
meanNoise=0; %mean noise for each fourier series
y=zeros(6,15); %final output - prediction values
noisyRegLow=zeros(1,15); %Vector representing low end of prediction band
(linear regression - noise)
noisyRegHigh=zeros(1,15); %Vector representing upper end of prediction band
(linear regression + noise)


t=[1:1:N+15];      %Time Index: N-t days before present day
p=polyfit(t(1:N)',x,1);         %Perform least-squares linear regression
regression=p(1)*t+p(2);    %Calculate Linear Regression from polyfit data
difference=x'-regression(1:N);  %Subtract regression from Actual
fourier2=fit(t(1:N)',difference','Fourier2'); %Perform Fourier2 Fit on
difference
fourier=feval(fourier2,t);  %Evaluate Fourier Series for all T



 noise=difference'-fourier(1:N);
  meanNoise= (max(abs(noise))- min(abs(noise)))/2;
  figure(1);
  fourievaluated=regression'+fourier;
  noisyRegLow(1:15)=fourievaluated((N+1):(N+15))-meanNoise;
```

```
  noisyRegHigh(1:15)=fourievaluated((N+1):(N+15))+meanNoise;
  subplot(2,1,1);
  plot(t,fourievaluated);
  hold on;
  plot([N+1:N+15],noisyRegLow,'black');
  plot([N+1:N+15],noisyRegHigh,'black');
  legend('Prediction', 'Noise Band', 'Location', 'North');
  scatter(t(1:N),x','o');
  scatter([N+1:1:N+15],r,'*','r');
  xlabel('Market Days');
  ylabel('Daily Closing Price');
  plotTitle=sprintf('Linear Regression Stock Price Prediction with FourierTwo
And Noise Band for %s', s);
  title(plotTitle);
  y(1,:)=fourievaluated(N+1:N+15);
  z(1,:)=fourievaluated;
  error=abs((r-y(1,:))./r)*100;
  subplot(2,1,2);
  scatter([1:1:15],error,'*');
  title('Error Associated with Prediction');
  xlabel('Market Days From September 15 to October 10');
  ylabel('Percent Error');

% ARIMA Model
  Mdl=arima(2,1,2);  %Arima 2,1,2 model
  EstMdl=estimate(Mdl,x); %Estimate Arima Parameters from input data
  [arimaPredictions arimaErr]=forecast(EstMdl,15,'Y0',x); %Forecast future
prices
  [Resids Varies] = infer(EstMdl,x); %Infer past results
  arimaplot = [(Resids+x)' arimaPredictions'];
  z(2,:)=arimaplot;
  y(2,:)=z(2,end-14:end);

  %Plot ARIMA Results
  figure(2);
  subplot(2,1,1);
  hold on;
  scatter(t(1:N),x,'b','o');
  plot(1:N+15,arimaplot);
  scatter([N+1:N+15],r,'r','*');
  legend('Past Prices','Predicted Price', 'Actual Future Price');
  noisyRegLow(1:15)=z(2,(N+1):(N+15))-meanNoise;
  noisyRegHigh(1:15)=z(2,(N+1):(N+15))+meanNoise;
  plot([N+1:N+15],noisyRegLow,'black');
  plot([N+1:N+15],noisyRegHigh,'black');
  plotTitle=sprintf('Stock Price Predictions for %s using ARIMA(2,1,2)
Model',s);
  title(plotTitle);
  xlabel('Market Days');
  ylabel('Daily Closing Price');
  subplot(2,1,2);
  error= abs((r-arimaPredictions')./r)*100;
  scatter([1:1:15],error,'*');
  title('Error Associated with Prediction');
  xlabel('Market Days From September 15 to October 11');
  ylabel('Percent Error');
```

```
  %Fourier2 + Arima(2,1,2) Model

 fourier2=fit(t(1:N)',x,'Fourier2'); %Perform Fourier2 Fit on x
 fourier=feval(fourier2,t);  %Evaluate Fourier Series for all T
 residuals=x-fourier(1:N);

  Mdl=arima(2,1,2);  %Arima 2,1,2 model
  EstMdl=estimate(Mdl,residuals); %Estimate Arima Parameters from input data
  [arimaPredictions arimaErr]=forecast(EstMdl,15,'Y0',residuals); %Forecast
future prices
  [Resids Varies] = infer(EstMdl,residuals); %Infer past results
  arimaplot = [(Resids+residuals)' arimaPredictions'];
  arimaFourierPlot = arimaplot+fourier';
  z(3,:)=arimaFourierPlot;
  y(3,:)=z(3,end-14:end);

  %Plot ARIMA+Fourier Results

  figure(3);
  subplot(2,1,1);
  hold on;
  scatter(t(1:N),x,'b','o');
  plot(1:N+15,arimaFourierPlot);
  scatter([N+1:N+15],r,'r','*');
  legend('Past Prices','Predicted Price', 'Actual Future Price');
  noisyRegLow(1:15)=z(3,(N+1):(N+15))-meanNoise;
  noisyRegHigh(1:15)=z(3,(N+1):(N+15))+meanNoise;
  plot([N+1:N+15],noisyRegLow,'black');
  plot([N+1:N+15],noisyRegHigh,'black');
  plotTitle=sprintf('Stock Price Predictions for %s using Fourier2 and
ARIMA(2,1,2) Model',s);
  title(plotTitle);
  xlabel('Market Days');
  ylabel('Daily Closing Price');
  subplot(2,1,2);
  error= abs(((r-y(3,:)))./r)*100;
  scatter([1:1:15],error,'*');
  title('Error Associated with Prediction');
  xlabel('Market Days From September 15 to October 11');
  ylabel('Percent Error');


  %Stock Exchange Index Model
 ex_x= exchange;
 ex_N= length(ex_x);
 ex_t=[1:1:ex_N+15];              %Time Index: N-t days before present day
 ex_p=polyfit(ex_t(1:ex_N)',ex_x,1);      %Perform least-squares linear
regression
 regression=ex_p(1)*ex_t+ex_p(2);         %Calculate Linear Regression from
polyfit data
 difference=ex_x'-regression(1:ex_N);  %Subtract regression from Actual
 fourier2=fit(ex_t(1:ex_N)',difference','Fourier2'); %Perform Fourier2 Fit on
difference
 fourier=feval(fourier2,ex_t);      %Evaluate Fourier Series for all T
```

```
 ex_noise=difference'-fourier(1:ex_N);
 ex_meanNoise= (max(abs(ex_noise)) - min(abs(ex_noise)))/2;

 ex_fourievaluated=regression+fourier';  %Model of Stock Exchange for one
period + 15 days

 %Plot Exchange Index Model
 %figure(4);
 %plot(ex_t,ex_fourievaluated);
 %hold on;
 %scatter(1:ex_N,ex_x);
 %xlabel('Market Days');
 %ylabel('Daily Closing Index Value (Actual and Modeled)');
 %title('FourierTwo and Linear Regression Model for Stock Exchange');

 %Scale down exchange to magnitude of stock price
% scale_Factor = mean(ex_x)/mean(x);   %Scale by means
  scale_Factor = ex_x(ex_N)/x(N); %Scale so two signals are equal on
september 12
 ex_scaled = ex_fourievaluated/scale_Factor;


 %Fit the two models to the same vector length

 if(N+15>ex_N+15)    %Stock period is longer than exchange period
     nan=NaN(N-ex_N);
     ex_fitted = [nan(1,:) ex_scaled]; %Zero-pad beginning of exchange model
 elseif(ex_N+15 > N+15)  %Exchange period is longer than stock period
     ex_fitted = ex_scaled((ex_N-N+1):ex_N+15);  %Shorten by removing past
values
 else    %Two periods are equal
     ex_fitted = ex_scaled;
 end


 %Determine Security Volatility = Standard Deviation
x_volatility = sqrt(sum((x-mean(x)).^2))/mean(x);
ex_volatility = sqrt(sum((ex_x-mean(ex_x)).^2))/mean(ex_x);

 v_ratio = x_volatility/(ex_volatility+x_volatility);
 v_ratio_n= 1-v_ratio;

 %Integrate Exchange Model into Stock Models

 %Fourier2 and Linear Model Plus Exchange Predictions
 z(4,:) = (v_ratio*ex_fitted)+(v_ratio_n*z(1,:));
 y(4,:) = z(4,end-14:end);
 %Arima(2,1,2) Model Plus Exchange Predictions
 z(5,:) = (v_ratio*ex_fitted)+(v_ratio_n*z(2,:));
 y(5,:) = z(5,end-14:end);
 %Fourier2+ARIMA(2,1,2) Model plus Exchange Predictions
 z(6,:) = (v_ratio*ex_fitted)+(v_ratio_n*z(3,:));
 y(6,:) = z(6,end-14:end);
```

```
%Plot Fourier2 and Linear Model Plus Exchange Results
 figure(5);
 noisyRegLow(1:15)=z(4,(N+1):(N+15))-meanNoise;
 noisyRegHigh(1:15)=z(4,(N+1):(N+15))+meanNoise;
 subplot(2,1,1);
 plot(t,z(4,:));
 hold on;
 plot([N+1:N+15],noisyRegLow,'black');
 plot([N+1:N+15],noisyRegHigh,'black');
 legend('Prediction', 'Noise Band', 'Location', 'North');
 scatter(t(1:N),x','o');
 scatter([N+1:1:N+15],r,'*','r');
 xlabel('Market Days');
 ylabel('Daily Closing Price');
 plotTitle=sprintf('Linear Regression and Fourier2 Plus Exchange Index for
%s', s);
 title(plotTitle);

 error=abs((r-y(4,:))./r)*100;
 subplot(2,1,2);
 scatter([1:1:15],error,'*');
 title('Error Associated with Prediction');
 xlabel('Market Days From September 15 to October 10');
 ylabel('Percent Error');


%Plot Arima(2,1,2) Model Plus Exchange Results

figure(6);
 noisyRegLow(1:15)=z(5,(N+1):(N+15))-meanNoise;
 noisyRegHigh(1:15)=z(5,(N+1):(N+15))+meanNoise;
 subplot(2,1,1);
 plot(t,z(5,:));
 hold on;
 plot([N+1:N+15],noisyRegLow,'black');
 plot([N+1:N+15],noisyRegHigh,'black');
 legend('Prediction', 'Noise Band', 'Location', 'North');
 scatter(t(1:N),x','o');
 scatter([N+1:1:N+15],r,'*','r');
 xlabel('Market Days');
 ylabel('Daily Closing Price');
 plotTitle=sprintf('ARIMA(2,1,2) Plus Exchange Index for %s', s);
 title(plotTitle);

 error=abs((r-y(5,:))./r)*100;
 subplot(2,1,2);
 scatter([1:1:15],error,'*');
 title('Error Associated with Prediction');
 xlabel('Market Days From September 15 to October 10');
 ylabel('Percent Error');

%Plot Fourier2 and ARIMA(2,1,2) Model plus Exchange Results

figure(7);
 noisyRegLow(1:15)=z(6,(N+1):(N+15))-meanNoise;
```

```
  noisyRegHigh(1:15)=z(6,(N+1):(N+15))+meanNoise;
  subplot(2,1,1);
  plot(t,z(6,:));
  hold on;
  plot([N+1:N+15],noisyRegLow,'black');
  plot([N+1:N+15],noisyRegHigh,'black');
  legend('Prediction', 'Noise Band', 'Location', 'North');
  scatter(t(1:N),x','o');
  scatter([N+1:1:N+15],r,'*','r');
  xlabel('Market Days');
  ylabel('Daily Closing Price');
  plotTitle=sprintf('Fourier2 and ARIMA(2,1,2) Plus Exchange Index for %s',
s);
  title(plotTitle);

  error=abs((r-y(6,:))./r)*100;
  subplot(2,1,2);
  scatter([1:1:15],error,'*');
  title('Error Associated with Prediction');
  xlabel('Market Days From September 15 to October 10');
  ylabel('Percent Error');



end
```

Jacob Grotton

```
% Jacob Grotton


function [model,model2] = IQPModel1(stock, autoc, name, fstocks)
future = length(fstocks);

list = stock(end-(autoc-1): end);
time = transpose(1:autoc);
mtime = transpose(1: autoc + future);
ftime = transpose(autoc + 1: autoc + future);
utime = zeros(1, length(mtime));
linf = fit(time, list, 'poly1');
linv = linf(time);

fourf2 = fit(time, list, 'fourier2');
fourv2 = fourf2(time);

error = list - linv;
error2 = list - fourv2;

fourf = fit(time, error, 'fourier2');
fourv = fourf(time);

linf2 = fit(time, error, 'poly1');
linv2 = linf2(time);
```

```
noise = fourv - error;
noise2 = linv2 - error2;

t = [name ' Model - Linear Fourier2'];
q = [name ' Model Error - Linear Fourier2'];
t2 = [name ' Model -Fourier2  Linear'];
q2 = [name ' Model Error -Fourier2 Linear'];

model = fourf(mtime) + linf(mtime);
model2 = fourf2(mtime)+linf2(mtime);

errorup = model + max(noise);
errordown = model + min(noise);
errorup2 = model2 + max(noise2);
errordown2 = model2 + min(noise2);
fullstock = [transpose(list) transpose(fstocks)];
fullstock = transpose(fullstock);

ferror = fullstock - model;
ferror2 = fullstock - model2;


figure;
subplot(2,2,1);
scatter(time, list,'.','blue')
title(t)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model,'Color','black')
plot(errorup,':','Color','black')
plot(errordown,':','Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,3);
scatter(time,ferror(1:end-future),'.','blue')
title(q)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror(end-future+1:end),'.','red')
plot(max(noise)+utime,':','Color','black')
plot(min(noise)+utime,':','Color','black')
hold off

subplot(2,2,2);
scatter(time, list,'.','blue')
title(t2)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model2,'Color','black')
plot(errorup2,':','Color','black')
```

```
plot(errordown2,':','Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,4);
scatter(time,ferror2(1:end-future),'.','blue')
title(q2)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror2(end-future+1:end),'.','red')
plot(max(noise2)+utime,':','Color','black')
plot(min(noise2)+utime,':','Color','black')
hold off

end

function [model,model2] = IQPModel2(stock, autoc, name, fstocks)
future = length(fstocks);

list = stock(end-(autoc-1): end);
time = transpose(1:autoc);
mtime = transpose(1: autoc + future);
ftime = transpose(autoc + 1: autoc + future);
utime = zeros(1, length(mtime));
linf = fit(time, list, 'poly1');
linv = linf(time);

mdl = arima(1,1,1);

estmdl = estimate(mdl, list);
[res,var] = infer(estmdl, list);
arimM = res + list;
[fut,err] = forecast(estmdl,future,'Y0', list);
model = [transpose(arimM) transpose(fut)];
model = model';

error = list - linv;

mdl2 = arima(2,1,2);

estmdl2 = estimate(mdl2, error);
[res2,var2] = infer(estmdl2, error);
arimM2 = res2 + list;
[fut2,err2] = forecast(estmdl2,future,'Y0', error);
finMod2 = arimM2 + error;
fut2 = fut2 + mean(arimM2);
model2 = [transpose(finMod2) transpose(fut2)];
model2 = model2';

t = [name ' Model - ARIMA [1,1,1]'];
q = [name ' Model Error - ARIMA [1,1,1]'];
t2 = [name ' Model - Linear ARIMA[1,1,1]'];
q2 = [name ' Model Error - Linear ARIMA[1,1,1]'];
```

```
fullstock = [transpose(list) transpose(fstocks)];
fullstock = transpose(fullstock);

ferror = fullstock - model;
ferror2 = fullstock - model2;



figure;
subplot(2,2,2);
scatter(time, list,'.','blue')
title(t2)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model2,'Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,4);
scatter(time,ferror2(1:end-future),'.','blue')
title(q2)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror2(end-future+1:end),'.','red')
hold off

subplot(2,2,1);
scatter(time, list,'.','blue')
title(t)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model2,'Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,3);
scatter(time,ferror(1:end-future),'.','blue')
title(q)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror(end-future+1:end),'.','red')
hold off


end
```

Term B

Sidney Veilleux

```
%Sidney Veilleux
%IQP Term B
%Week Seven Model for Stock Price Prediction
%This function consumes ten real-valued periodic signals one period in length
%A String representing Ten Stock Symbols
%A vector representing ten stocks * five weeks of actual values for
determining prediction accuracy
%A vector representing ten stocks * one period of Stock Exchange index values
%Calculates Normalized Volatility of each Stock Signal and determines which
model to use
%Creates Linear Regression + beta * FourierTwo Model for Low-Volatility
Stocks (v<0.3)
%Creates Linear Regression + FourierTwo Model Model for Mid-Volatility Stocks
(0.3<v<0.8)
%Creates FourierTwo+ARIMA Model for High-Volatility Stocks (v>0.8)

%Integrates exchange model into stock price model using three different
equations:
%Plots results of two resulting models along with prediction percent error
%Returns Vector y containing ndays days of predictions

function [average_average_error, average_max_error] =
weekSevenAnalysis(stocks,real_values,stock_names,exchange,ndays,alphaOne,alph
aTwo,alphaThree,time_shiftOne,time_shiftTwo,beta)
close all;

num_stocks = size(stocks);
num_stocks = num_stocks(2);
average_error = zeros(3,num_stocks);
max_error = zeros(3,num_stocks);

for e = [1:1:num_stocks]
    x = stocks(:,e);  %Current stock data (one autocorrelation period)
ex_x= exchange(:,e);
    r = real_values(e,:);
    s = stock_names(((e-1)*4)+1:(e)*4);

%Grab one autocorrelation period (vectors have trailing zeros)
    N = length(x);
    found=0;
for k=[1:1:N]
if(found == 0)
if(x(k)==0)
x_length=k-1;
           found=1;
end
end
end
    x=x(1:x_length);
    z = zeros(2,x_length+ndays);
%Remove trailing zeros from exchange data
```

```
ex_N = length(ex_x);
    found=0;
for k=[1:1:ex_N]
if(found == 0)
if(ex_x(k)==0)
ex_length=k-1;
           found=1;
end
end
end
ex_x=ex_x(1:ex_length);

%Remove trailing zeros from stock name
    found=0;
s_N=4;
s_length=4;
for k = 1:1:s_N;
if(found == 0)
if(s(k)=='0')
s_length=k-1;
           found=1;
end
end
end
    s=s(1:s_length);


    N=length(x);   %This is the autocorrelation value
%Populate some vectors which are iterated over in a for loop
    noise=zeros(1,N);   %noise vector for each fourier series
    y=zeros(2,ndays); %current stock pprediction values
    t=[1:1:N+ndays];                %Time Index: N-t days before present day
%Stock Exchange Index Model
ex_N= length(ex_x);
ex_t=[1:1:ex_N+ndays];             %Time Index: N-t days before present day

%Model Exchange Index with Second-Order Fourier Series and ARIMA(2,1,2)
    ex_fourier2=fit(ex_t(1:ex_N)',ex_x,'Fourier2'); %Perform Fourier2 Fit on
x
ex_fourier=feval(ex_fourier2,ex_t);  %Evaluate Fourier Series for all T
ex_residuals=ex_x-ex_fourier(1:ex_N);
ex_Mdl=arima(2,1,2);  %Arima 2,1,2 model
ex_EstMdl=estimate(ex_Mdl,ex_residuals); %Estimate Arima Parameters from
input data

[ex_arimaPredictionsex_arimaErr]=forecast(ex_EstMdl,ndays,'Y0',ex_residuals);
%Forecast future prices
    [ex_Residsex_Varies] = infer(ex_EstMdl,ex_residuals); %Infer past results
ex_arimaplot = [(ex_Resids+ex_residuals)' ex_arimaPredictions'] +
ex_fourier';

%Perform Linear Regression and Fourier to Determine Noise Band for Exchange
Model
ex_p=polyfit(ex_t(1:ex_N)',ex_x,1);      %Perform least-squares linear
regression
```

```
ex_regression=ex_p(1)*ex_t+ex_p(2);          %Calculate Linear Regression from
polyfit data
ex_difference=ex_x'-ex_regression(1:ex_N);  %Subtract regression from Actual
    ex_fourier2=fit(ex_t(1:ex_N)',ex_difference','Fourier2'); %Perform
Fourier2 Fit on difference
ex_fourier=feval(ex_fourier2,ex_t);      %Evaluate Fourier Series for all T
ex_noise=ex_difference'-ex_fourier(1:ex_N);
ex_meanNoise= (max(abs(ex_noise)) - min(abs(ex_noise)))/2;

ex_model=ex_arimaplot;  %Model of Stock Exchange for one period + ndays days

%Determine Security Volatility = Standard Deviation
x_volatility = sqrt(sum((x-mean(x)).^2))/mean(x);
ex_volatility = sqrt(sum((ex_x-mean(ex_x)).^2))/mean(ex_x);
v_ratio = x_volatility/(ex_volatility+x_volatility);
v_ratio_n= 1-v_ratio;


%Calcualte FourierTwo and Linreg to get scalar noise value for models
    p=polyfit(t(1:N)',x,1);         %Perform least-squares linear regressio
    regression=p(1)*t+p(2);         %Calculate Linear Regression from polyfit
data
    difference=x'-regression(1:N);  %Subtract regression from Actual
    fourier2=fit(t(1:N)',difference','Fourier2'); %Perform Fourier2 Fit on
difference
fourier=feval(fourier2,t);      %Evaluate Fourier Series for all T
%Calculate Noise
    noise=difference'-fourier(1:N);
meanNoise= (max(abs(noise))- min(abs(noise)))/2;



%LinReg + FourierTwo
fourievaluated=regression'+fourier;
        y(1,:)=fourievaluated(N+1:N+ndays);
        z(1,:)=fourievaluated';
plotTitle=sprintf('Linear Regression Stock Price Prediction with FourierTwo
And Noise Band for %s', s);



%Fourier2 + Arima(2,1,2) Model
        fourier2=fit(t(1:N)',x,'Fourier2'); %Perform Fourier2 Fit on x
fourier=feval(fourier2,t);  %Evaluate Fourier Series for all T
        residuals=x-fourier(1:N);

Mdl=arima(2,1,2);  %Arima 2,1,2 model
EstMdl=estimate(Mdl,residuals); %Estimate Arima Parameters from input data
        [arimaPredictionsarimaErr]=forecast(EstMdl,ndays,'Y0',residuals);
%Forecast future prices
        [Resids Varies] = infer(EstMdl,residuals); %Infer past results
arimaplot = [(Resids+residuals)' arimaPredictions'];
arimaFourierPlot = arimaplot+fourier';
        z(1,:)=z(1,:)+arimaFourierPlot;
        y(1,:)=z(1,end-(ndays-1):end);
plotTitle=sprintf('Stock Price Predictions for %s using Fourier2 and
ARIMA(2,1,2) Model',s);
```

```
noisyRegLow=y(1,:)./2-meanNoise; %Vector representing low end of prediction
band (linear regression - noise)
noisyRegHigh=y(1,:)./2+meanNoise; %Vector representing upper end of
prediction band (linear regression + noise)
%Plot Model Results and Error
    figure;
    subplot(2,1,1);
    plot(t,z(1,:)./2);
    hold on;
    plot([N+1:N+ndays],noisyRegLow,'black');
    plot([N+1:N+ndays],noisyRegHigh,'black');
    legend('Prediction', 'Noise Band', 'Location', 'North');
    scatter(t(1:N),x','o');
    scatter([N+1:1:N+ndays],r,'*','r');
xlabel('Market Days');
ylabel('Daily Closing Price (Actual and Predicted)');
    title(plotTitle);
    subplot(2,1,2);
    error=abs((r-(y(1,:)./2))./r)*100;
    scatter([1:1:ndays],error,'*');
    title('Error Associated with Prediction');
xlabel('Market Days From September 15 to November 7');
ylabel('Percent Error');


%Plot Exchange Index Model
% figure;
% plot(ex_t,ex_model);
% hold on;
%scatter(1:ex_N,ex_x);
% xlabel('Market Days');
% ylabel('Daily Closing Index Value (Actual and Modeled)');
% plottitle = sprintf('FourierTwo and Linear Regression Model for Stock
Exchange of %s', s);
% title(plottitle);

%Normalize exchange
scale_Factor = mean(ex_x)/mean(x);   %Scale by the means
%scale_Factor = ex_x(ex_N)/x(N); %Scale so two signals are equal on September
12
ex_scaled = ex_model/scale_Factor;


%Fit the two models to the same vector length
if(N>ex_N)    %Stock period is longer than exchange period
     nan=NaN(N-ex_N);
ex_scaled_fitted = [nan(1,:) ex_scaled]; %Zero-pad beginning of exchange
model with nans
ex_fitted = ex_x;  %Can't zero pad (or nan-pad) if correlation is being
calculated
fitted_length = ex_N;
elseif(ex_N> N)  %Exchange period is longer than stock period
```

```
ex_scaled_fitted = ex_scaled((ex_N-N+1):ex_N+ndays);  %Shorten by removing
past values
ex_fitted = ex_x((ex_N-N+1):ex_N);
fitted_length = N;
else%Two periods are equal
ex_scaled_fitted = ex_scaled;
ex_fitted = ex_x;
fitted_length = N;
end


%Integrate Exchange Model into Stock Models
%ex_fitted is exchange model
%z(1,:) is stock model
%z(2:4,:) is combined models 1-3
ex_scaled_fitted_shiftedOne = [zeros(1,time_shiftOne) ex_scaled_fitted(1:end-
time_shiftOne)]; %shift exchange model (delay by time_shift)
ex_scaled_fitted_shiftedTwo = [zeros(1,time_shiftTwo) ex_scaled_fitted(1:end-
time_shiftTwo)]; %shift exchange model (delay by time_shift)
%ex_fitted_shiftedOne = [zeros(1,time_shiftOne) ex_fitted(1:end-
time_shiftOne)']; %shift exchange model (delay by time_shift)
ex_fitted_shiftedTwo = [zeros(1,time_shiftTwo) ex_fitted(1:end-
time_shiftTwo)']; %shift exchange model (delay by time_shift)

    R=corrcoef(x(N-fitted_length+1:N),ex_fitted);       %Determine
Correlation between stock and exchange index raw data
    correlation = R(2,1);          %Grab r-value (correlation coefficient)

    R=corrcoef(x(N-fitted_length+1:N),ex_fitted_shiftedTwo);       %Determine
Correlation between stock and exchange index raw data
correlation_shifted = R(2,1);          %Grab r-value (correlation
coefficient)

%Three Equations to Combine Stock and Exchange Models

%Equation One:    Xs[n] +  aXe[n-k]   a is alpha(1), k is time-shift
     z(2,:) = z(1,:)./2 + alphaOne*ex_scaled_fitted_shiftedOne;  %Combine
stock and exchange
     y(2,:) = z(2,end-(ndays-1):end);
%Equation Two:    Xs[n] + arXe[n-k]   a is alpha(2), k is time-shift
     z(3,:) = z(1,:)./2 +
alphaTwo.*correlation_shifted*ex_scaled_fitted_shiftedTwo;  %Combine stock
and exchange
    y(3,:) = z(3,end-(ndays-1):end);
%Equation Three: bXs[n] + arXe[n]     a is alpha(3), b is beta
     z(4,:) = beta.*z(1,:)./2 + alphaThree.*correlation*ex_scaled_fitted;
%Combine stock and exchange
    y(4,:) = z(4,end-(ndays-1):end);



%Plot Model Plus Exchange Results
for k=1:1:3

        figure;
```

```
noisyRegLow(1:ndays)=z(k+1,(N+1):(N+ndays))-meanNoise;
noisyRegHigh(1:ndays)=z(k+1,(N+1):(N+ndays))+meanNoise;
        subplot(2,1,1);
        plot(t,z(k+1,:));
        hold on;
        plot([N+1:N+ndays],noisyRegLow,'black');
        plot([N+1:N+ndays],noisyRegHigh,'black');
        legend('Prediction', 'Noise Band', 'Location', 'North');
        scatter(t(1:N),x','o');
        scatter([N+1:1:N+ndays],r,'*','r');
xlabel('Market Days');
ylabel('Daily Closing Price (Actual and Predicted)');
plotTitle=sprintf('Stock Price Model with Exchange Index for %s using
Equation %d', s,k);
        title(plotTitle);

        error=abs((r-y(k+1,:))./r)*100;
average_error(k,e) = mean(error);
max_error(k,e) = max(error);
        subplot(2,1,2);
        scatter([1:1:ndays],error,'*');
        title('Error Associated with Prediction');
xlabel('Market Days From September 15 to November 7');
ylabel('Percent Error');
end
end
average_average_error = zeros(1,3);
average_max_error = zeros(1,3);
for k=1:1:3
average_average_error(k) = mean(average_error(k,:));
average_max_error(k) = mean(max_error(k,:));
end
end
```

Jacob Grotton

```
function [model,model2] = IQPModel4(stock, autoc, name, fstocks,
exchange,E_autoc)
future = length(fstocks);

list = stock(end-(autoc-1): end);
time = transpose(1:autoc);
mtime = transpose(1: autoc + future);
ftime = transpose(autoc + 1: autoc + future);
utime = zeros(1, length(mtime));
linf = fit(time, list, 'poly1');
linv = linf(time);

error = list - linv;

fourf = fit(time, error, 'fourier2');
fourv = fourf(time);
```

```
noise = fourv - error;

Elist = exchange(end-(E_autoc-1): end);
Etime = transpose(1:E_autoc);
Emtime = transpose(1: E_autoc + future);
Eftime = transpose(E_autoc + 1: E_autoc + future);
utime = zeros(1, length(Emtime));
Elinf = fit(Etime, Elist, 'poly1');
Elinv = Elinf(Etime);

Eerror = Elist - Elinv;

Efourf = fit(Etime, Eerror, 'fourier2');
Efourv = Efourf(Etime);

% Enoise = Efourv - Eerror;

mdl = arima(1,1,1);

estmdl = estimate(mdl, list);
[res,var] = infer(estmdl, list);
amodel = res + list;
[fut,err] = forecast(estmdl,future,'Y0', list);

Eestmdl = estimate(mdl, Elist);
[Eres,Evar] = infer(Eestmdl, Elist);
Eamodel = Eres + Elist;
[Efut,Eerr] = forecast(Eestmdl,future,'Y0', Elist);


t = [name ' Model - Linear Fourier2 w/ Exchange(20-Day)'];
q = [name ' Model Error - Linear Fourier2 w/ Exchange'];
t2 = [name ' Model - ARIMA [1,1,1] w/ Exchange(20-Day)'];
q2 = [name ' Model Error - ARIMA [1,1,1] w/ Exchange'];

vol = sqrt(mean(list-mean(list)).^2);
Evol = sqrt(mean(list-mean(Elist)).^2);

pmodel = fourf(time) + linf(time);
Epmodel = Efourf(Etime) + Elinf(Etime);

nmodel = fourf(ftime)+linf(ftime);
Enmodel = Efourf(Eftime) + Elinf(Eftime);

Escale = mean(list)/mean(Elist);

fmodel = (vol/(vol+Evol))*nmodel + (Evol/(vol+Evol))*Escale*Enmodel;

model = [pmodel' fmodel'];
model = model';
afut = (vol/(vol+Evol))*fut + (Evol/(vol+Evol))*Escale*Efut;
```

```
model2 = [transpose(amodel) transpose(afut)];
model2 = model2';


errorup = model + max(noise);
errordown = model + min(noise);
fullstock = [transpose(list) transpose(fstocks)];
fullstock = transpose(fullstock);

ferror = fullstock - model;
ferror2 = fullstock - model2;

figure;
subplot(2,2,1);
scatter(time, list,'.','blue')
title(t)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model,'Color','black')
plot(errorup,':','Color','black')
plot(errordown,':','Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,3);
scatter(time,ferror(1:end-future),'.','blue')
title(q)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror(end-future+1:end),'.','red')
plot(max(noise)+utime,':','Color','black')
plot(min(noise)+utime,':','Color','black')
hold off

subplot(2,2,2);
scatter(time, list,'.','blue')
title(t2)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model2,'Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,4);
scatter(time,ferror2(1:end-future),'.','blue')
title(q2)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror2(end-future+1:end),'.','red')
hold off
```

```
end


function [error1,error2] = IQPModel5(stock, autoc, name, fstocks,
exchange,E_autoc,vol)
future = length(fstocks);

list = stock(end-(autoc-1): end);
time = transpose(1:autoc);
mtime = transpose(1: autoc + future);
ftime = transpose(autoc + 1: autoc + future);
utime = zeros(1, length(mtime));
linf = fit(time, list, 'poly1');
linv = linf(time);

error = list - linv;

fourf = fit(time, error, 'fourier2');
fourv = fourf(time);

noise = fourv - error;

Elist = exchange(end-(E_autoc-1): end);
Etime = transpose(1:E_autoc);
Emtime = transpose(1: E_autoc + future);
Eftime = transpose(E_autoc + 1: E_autoc + future);
utime = zeros(1, length(Emtime));
Elinf = fit(Etime, Elist, 'poly1');
Elinv = Elinf(Etime);

Eerror = Elist - Elinv;

Efourf = fit(Etime, Eerror, 'fourier2');
Efourv = Efourf(Etime);

% Enoise = Efourv - Eerror;

mdl = arima(1,1,1);

estmdl = estimate(mdl, list);
[res,var] = infer(estmdl, list);
amodel = res + list;
[fut,err] = forecast(estmdl,future,'Y0', list);

Eestmdl = estimate(mdl, Elist);
[Eres,Evar] = infer(Eestmdl, Elist);
Eamodel = Eres + Elist;
[Efut,Eerr] = forecast(Eestmdl,future,'Y0', Elist);



t = [name ' Model - Linear Fourier2 w/ Volume Exchange(35-Day)'];
q = [name ' Model Error - Linear Fourier2 w/ Volume Exchange'];
```

```
t2 = [name ' Model - ARIMA [1,1,1] w/ Volume Exchange(35-Day)'];
q2 = [name ' Model Error - ARIMA [1,1,1] w/ Volume Exchange'];

rat = -0.000000000000032*(vol-3000000)^2+.03;

pmodel = fourf(time) + linf(time);
Epmodel = Efourf(Etime) + Elinf(Etime);

nmodel = fourf(ftime)+linf(ftime);
Enmodel = Efourf(Eftime) + Elinf(Eftime);

Escale = list(end)/Elist(end);

fmodel = (1-rat)*nmodel + rat*Escale*Enmodel;

model = [pmodel' fmodel'];
model = model';
afut = (1-rat)*fut + rat*Escale*Efut;


model2 = [transpose(amodel) transpose(afut)];
model2 = model2';


errorup = model + max(noise);
errordown = model + min(noise);
fullstock = [transpose(list) transpose(fstocks)];
fullstock = transpose(fullstock);

ferror = fullstock - model;
error1 = mean(abs(ferror)/mean(fullstock));
ferror2 = fullstock - model2;
error2 = mean(abs(ferror2)/mean(fullstock));



figure;
subplot(2,2,1);
scatter(time, list,'.','blue')
title(t)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model,'Color','black')
plot(errorup,':','Color','black')
plot(errordown,':','Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,3);
scatter(time,ferror(1:end-future),'.','blue')
title(q)
xlabel('Time (Days)')
ylabel('Error')
```

```
hold on
scatter(ftime,ferror(end-future+1:end),'.','red')
plot(max(noise)+utime,':','Color','black')
plot(min(noise)+utime,':','Color','black')
hold off

subplot(2,2,2);
scatter(time, list,'.','blue')
title(t2)
xlabel('Time (Days)')
ylabel('Price ($)')
hold on
plot(model2,'Color','black')
scatter(ftime, fstocks,'.','red')
hold off

subplot(2,2,4);
scatter(time,ferror2(1:end-future),'.','blue')
title(q2)
xlabel('Time (Days)')
ylabel('Error')
hold on
scatter(ftime,ferror2(end-future+1:end),'.','red')
hold off

end
```

Term C

Sidney Veilleux

```
%Sidney Veilleux
%MH IQP 2014-2015
%Final Model for Stock Price Prediction
%This function consumes:
 %An arbitrary number n of real-valued periodic stock signals, one period in
length
    %In the form of a 200-by-n matrix (append zeros so each signal is of
length 200)
 %A String representing all the Stock Symbols, each four chars in length
(append zeros) (ex 'A000ENE0AMD0AMZN')
 %A vector of length n containing quantities of each stock purchased (for
calculating profit)
 %A vector of length n containing values (1) or (-1) representing whether
each stock was bought (1) or short-sold (-1)
 %A 200-by-n matrix representing one period of Stock Exchange index values
for each stock
 %Constant coefficient alpha (recommend 0.1)
 %Constant coefficient beta (recommend 0.4)
 %An n-by-x matrix representing x future days of real stock prices (used to
check accuracy and calculate profit along with buy_sell and stock_quantities)
 %An x-length vector representing x future days of real exchange index values
(for plotting against portfolio total profit)
```

```
 %Calculates Normalized Volatility of each Stock Signal and determines which
model to use
        %Creates Linear Regression + beta * FourierTwo Model for Low-
Volatility Stocks (v<0.3)
        %Creates Linear Regression + FourierTwo Model Model for Mid-
Volatility Stocks (0.3<v<0.8)
        %Creates FourierTwo+ARIMA Model for High-Volatility Stocks (v>0.8)

%Integrates exchange model into stock price model
%Plots results of two resulting models along with prediction percent error
%Returns n-by-x matrix containing total profit
%Produces graphs of one month of predictions for each stock, with and without
exchange factored into model
%Produces graph of total portfoio profit plotted against aggregate market
performance

%Prints advice on whether to sell off stock midway through investing period
using percent gain/loss threshold


function [total_profit] =
OneMonthChallenge(stocks,stock_names,stock_quantities,buy_sell,exchange,alpha
,beta,real,exchange_real)
close all;

real_N = size(real);
real_N = real_N(2); %real_N is number of days
num_stocks = size(stocks);
num_stocks = num_stocks(2);
total_profit = zeros(num_stocks,real_N);  % Total profit acumulator
month_predictions=zeros(num_stocks,2);
all_predictions = zeros(num_stocks,24);

for e = [1:1:num_stocks]
    quantity = stock_quantities(e);
    b_s = buy_sell(e);
    x = stocks(:,e);  %Current stock data (one autocorrelation period)
    r = real(e,:);    %Current stock real "future" values
    ex_x= exchange(:,e);
    s = stock_names(((e-1)*4)+1:(e)*4);
    %Grab one autocorrelation period (vectors have trailing zeros)
    N = length(x);
    found=0;
    for k=[1:1:N]
       if(found == 0)
        if(x(k)==0)
           x_length=k-1;
           found=1;
        end
       end
    end
    x=x(1:x_length);
    for k=[1:1:real_N]
        total_profit(e,k) = (r(k)-x(x_length))*quantity*b_s;
    end
```

```
    z = zeros(2,x_length+24);
    %Remove trailing zeros from exchange data
     ex_N = length(ex_x);
    found=0;
    for k=[1:1:ex_N]
       if(found == 0)
        if(ex_x(k)==0)
           ex_length=k-1;
           found=1;
        end
       end
    end
    ex_x=ex_x(1:ex_length);

    %Remove trailing zeros from stock name
    found=0;
    s_N=4;
    s_length=4;
    for k = 1:1:s_N;
        if(found == 0)
        if(s(k)=='0')
           s_length=k-1;
           found=1;
        end
       end
    end
    s=s(1:s_length);


    N=length(x);   %This is the autocorrelation value
    %Populate some vectors which are iterated over in a for loop
    noise=zeros(1,N);   %noise vector for each fourier series
    y=zeros(2,24); %current stock ppediction values
    t=[1:1:N+24];                %Time Index: N-t days before present day
    %Stock Exchange Index Model
    ex_N= length(ex_x);
    ex_t=[1:1:ex_N+24];             %Time Index: N-t days before present day

    %Model Exchange Index with Second-Order Fourier Series and ARIMA(2,1,2)
    ex_fourier2=fit(ex_t(1:ex_N)',ex_x,'Fourier2'); %Perform Fourier2 Fit on
x
    ex_fourier=feval(ex_fourier2,ex_t);  %Evaluate Fourier Series for all T
    ex_residuals=ex_x-ex_fourier(1:ex_N);
    ex_Mdl=arima(2,1,2);  %Arima 2,1,2 model
    ex_EstMdl=estimate(ex_Mdl,ex_residuals); %Estimate Arima Parameters from
input data
    [ex_arimaPredictions
ex_arimaErr]=forecast(ex_EstMdl,24,'Y0',ex_residuals); %Forecast future
prices
    [ex_Resids ex_Varies] = infer(ex_EstMdl,ex_residuals); %Infer past
results
    ex_arimaplot = [(ex_Resids+ex_residuals)' ex_arimaPredictions'] +
ex_fourier';
```

```
    %Perform Linear Regression and Fourier to Determine Noise Band for
Exchange Model
    ex_p=polyfit(ex_t(1:ex_N)',ex_x,1);      %Perform least-squares linear
regression
    ex_regression=ex_p(1)*ex_t+ex_p(2);         %Calculate Linear Regression
from polyfit data
    ex_difference=ex_x'-ex_regression(1:ex_N);  %Subtract regression from
Actual
    ex_fourier2=fit(ex_t(1:ex_N)',ex_difference','Fourier2'); %Perform
Fourier2 Fit on difference
    ex_fourier=feval(ex_fourier2,ex_t);      %Evaluate Fourier Series for all
T
    ex_noise=ex_difference'-ex_fourier(1:ex_N);
    ex_meanNoise= (max(abs(ex_noise)) - min(abs(ex_noise)))/2;

    ex_model=ex_arimaplot;  %Model of Stock Exchange for one period + 24 days

    %Determine Security Volatility = Standard Deviation
    x_volatility = sqrt(sum((x-mean(x)).^2))/mean(x);
    ex_volatility = sqrt(sum((ex_x-mean(ex_x)).^2))/mean(ex_x);
    v_ratio = x_volatility/(ex_volatility+x_volatility);
    v_ratio_n= 1-v_ratio;


     %Calcualte FourierTwo and Linreg to get scalar noise value for models
    p=polyfit(t(1:N)',x,1);         %Perform least-squares linear regressio
    regression=p(1)*t+p(2);         %Calculate Linear Regression from polyfit
data
    difference=x'-regression(1:N);  %Subtract regression from Actual
    fourier2=fit(t(1:N)',difference','Fourier2'); %Perform Fourier2 Fit on
difference
    fourier=feval(fourier2,t);      %Evaluate Fourier Series for all T
    %Calculate Noise
    noise=difference'-fourier(1:N);
    meanNoise= (max(abs(noise))- min(abs(noise)))/2;

    if(x_volatility <0.35)  %Lowest Volatility Stocks Modeled with Linear
Regression, FourierTwo at Half Amplitude
        fourievaluated=regression'+fourier*beta;   %Scale-Down Fourier
Results
        y(1,:)=fourievaluated(N+1:N+24);
        z(1,:)=fourievaluated;
        plotTitle=sprintf('Linear Regression Stock Price Prediction with
FourierTwo And Noise Band for %s', s);
   elseif(x_volatility<0.8)  %Mid-range Volatility Stocks Modeled Using
LinReg + FourierTwo
        fourievaluated=regression'+fourier;
        y(1,:)=fourievaluated(N+1:N+24);
        z(1,:)=fourievaluated';
        plotTitle=sprintf('Linear Regression Stock Price Prediction with
FourierTwo And Noise Band for %s', s);

    else %Normalized Volatility above 0.8, Use FourierTwo + ARIMA
         %Fourier2 + Arima(2,1,2) Model
```

```
        fourier2=fit(t(1:N)',x,'Fourier2'); %Perform Fourier2 Fit on x
        fourier=feval(fourier2,t);  %Evaluate Fourier Series for all T
        residuals=x-fourier(1:N);

        Mdl=arima(2,1,2);  %Arima 2,1,2 model
        EstMdl=estimate(Mdl,residuals); %Estimate Arima Parameters from input
data
        [arimaPredictions arimaErr]=forecast(EstMdl,24,'Y0',residuals);
%Forecast future prices
        [Resids Varies] = infer(EstMdl,residuals); %Infer past results
        arimaplot = [(Resids+residuals)' arimaPredictions'];
        arimaFourierPlot = arimaplot+fourier';
        z(1,:)=arimaFourierPlot;
        y(1,:)=z(1,end-23:end);
        plotTitle=sprintf('Stock Price Predictions for %s using Fourier2 and
ARIMA(2,1,2) Model',s);
    end
    noisyRegLow=y(1,:)-meanNoise; %Vector representing low end of prediction
band (linear regression - noise)
    noisyRegHigh=y(1,:)+meanNoise; %Vector representing upper end of
prediction band (linear regression + noise)
    %Plot Model Results and Error
    figure;
    subplot(2,1,1);
    plot(t,z(1,:));
    hold on;
    plot([N+1:N+24],noisyRegLow,'black');
    plot([N+1:N+24],noisyRegHigh,'black');
    legend('Prediction', 'Noise Band', 'Location', 'North');
    scatter(t(1:N),x','o');
    scatter([N+1:N+real_N],r,'*','r');
    xlabel('Market Days');
    ylabel('Daily Closing Price');
    title(plotTitle);
    subplot(2,1,2);
    error=abs(100*(r-z(1,N+1:N+real_N))./r);
    scatter(1:real_N, error);
    xlabel('Market Days Starting January 26');
    ylabel('Prediction Percent Error');


%     %Plot Exchange Index Model
%     figure;
%     plot(ex_t,ex_model);
%     hold on;
%     scatter(1:ex_N,ex_x);
%     xlabel('Market Days');
%     ylabel('Daily Closing Index Value (Actual and Modeled)');
%     plottitle = sprintf('FourierTwo and Linear Regression Model for Stock
Exchange of %s', s);
%     title(plottitle);

    %Normalize exchange
    scale_Factor = mean(ex_x);   %Scale by the mean
    %scale_Factor = ex_x(ex_N)/x(N); %Scale so two signals are equal on
September 12
```

```
    ex_scaled = ex_model/scale_Factor;


    %Fit the two models to the same vector length
    if(N>ex_N)    %Stock period is longer than exchange period
     nan=NaN(N-ex_N);
     ex_scaled_fitted = [nan(1,:) ex_scaled]; %Zero-pad beginning of exchange
model with nans
     ex_fitted = ex_x;  %Can't zero pad (or nan-pad) if correlation is being
calculated
     fitted_length = ex_N;
    elseif(ex_N > N)  %Exchange period is longer than stock period
     ex_scaled_fitted = ex_scaled((ex_N-N+1):ex_N+24);  %Shorten by removing
past values
     ex_fitted = ex_x((ex_N-N+1):ex_N);
     fitted_length = N;
    else    %Two periods are equal
     ex_scaled_fitted = ex_scaled;
     ex_fitted = ex_x;
     fitted_length = N;
    end

    %Integrate Exchange Model into Stock Models
    %ex_fitted is exchange model
    %z(1,:) is stock model
    %z(2,:) is combined models

   % z(2,:) = (v_ratio*ex_fitted)+alpha*(v_ratio_n*z(1,:));  Old Method Using
Volatility Ratio
    % z(2,:) = (ex_fitted.*alpha + (1-alpha).*z(1,:));       Old Method using
just alpha value
    R=corrcoef(x(N-fitted_length+1:N),ex_fitted);       %Determine
Correlation between stock and exchange index raw data
    correlation = R(2,1);          %Grab r-value (correlation coefficient)



   % z(2,:) = z(1,:) .* (1 + alpha*correlation*ex_scaled_fitted);  %Combine
stock and exchange using the Humi Equation
    z(2,:) = z(1,:) .* (1 + alpha*correlation*ex_scaled_fitted);  %Combine
stock and exchange using the Humi Equation
    y(2,:) = z(2,end-23:end);

    %Plot Model Plus Exchange Results
    figure;
    subplot(2,1,1);
    noisyRegLow(1:24)=z(2,(N+1):(N+24))-meanNoise;
    noisyRegHigh(1:24)=z(2,(N+1):(N+24))+meanNoise;
    plot(t,z(2,:));                     %Plot Model Curve
    hold on;
    plot([N+1:N+24],noisyRegLow,'black'); %Plot Error Band low
    plot([N+1:N+24],noisyRegHigh,'black');%Plot Error Band high
    legend('Prediction', 'Noise Band', 'Location', 'North');
    scatter(t(1:N),x','o');             %Plot Real past values
    scatter([N+1:N+real_N],r,'*','r');  %Plot Real Future Values
```

```
    xlabel('Market Days');
    ylabel('Daily Closing Price');
    plotTitle=sprintf('Stock Price Model with Exchange Index for %s', s);
    title(plotTitle);
    subplot(2,1,2);
    error=abs(100*(r-z(2,N+1:N+real_N))./r);
    scatter(1:real_N, error);
    xlabel('Market Days Starting January 26');
    ylabel('Prediction Percent Error');

    month_predictions(e,1)=y(1,24);  %Stock Model Prediciton Value for One
month (January 1, 2015)
    month_predictions(e,2)=y(2,24);  %Stock + Exchange Prediciton Value for
One month (January 1, 2015)
    all_predictions(e,:) = y(1,:);
end


profit = zeros(1,real_N+1);
for k=1:1:real_N
    profit(k+1) = sum(total_profit(:,k));  % Daily Profit Calculator
end
figure;
subplot(2,1,1);
plot(0:real_N,profit);
hold on;
plot(0:real_N,zeros(1,real_N+1),'black');
hold on;
for k=1:1:real_N
    if(profit(k+1)>=0)
        scatter(k,profit(k+1),'o','green');
    else
        scatter(k,profit(k+1),'o','red');
    end
end
    scatter(0:real_N,profit,'*','black');

title('Total Profit/Loss offset from $100,000 portfolio');
ylabel('Profit (Dollars)');
xlabel('Market Days Starting January 26');
subplot(2,1,2);
scatter(0:real_N, [ex_x(end) exchange_real],'*');
hold on;
exchange_zero(1:real_N+1) = ex_x(end);
plot(0:real_N, [ex_x(end) exchange_real]);
plot(0:real_N, exchange_zero);
title('NYSE Index');
ylabel('Index Value (Dollars)');
xlabel('Market Days Starting January 26');

%Monetary Threshold For Changing Investments
for e=[1:1:num_stocks]
     s = stock_names(((e-1)*4)+1:(e)*4);
     %Remove trailing zeros from stock name
    found=0;
    s_N=4;
```

```
    s_length=4;
    for k = 1:1:s_N;
        if(found == 0)
        if(s(k)=='0')
           s_length=k-1;
           found=1;
        end
      end
    end
    s=s(1:s_length)
    bs = buy_sell(e);
    r = real(e,real_N);

    p = all_predictions(e,real_N)
    outcome = 100*((r-p)*bs)/r; %Percent above or below prediction value
    if((outcome < -5) && (bs==-1))
        advice = sprintf('%s (short) has risen more than 5 percent above
predicted value, cut your losses!',s)
    elseif((outcome < -2) && (bs == -1))
           advice = sprintf('%s (short) has risen more than 2 percent above
predicted value, consider cutting your losses.',s)
    elseif((outcome > 5 ) && (bs == -1))
            advice = sprintf('%s (short) has fallen more than 5 percent below
predicted value, cash in now!',s)
    elseif((outcome > 2 ) && (bs == -1))
            advice = sprintf('%s (short) has fallen more than 2 percent below
predicted value, consider cashing in',s)
    elseif((outcome < -5) && (bs == 1))
           advice = sprintf('%s (long) has fallen more than 5 percent below
predicted value, cut your losses!',s)
    elseif((outcome < -2) && (bs == 1))
           advice = sprintf('%s (long) has fallen more than 2 percent below
predicted value, consider cutting your losses.',s)
    elseif((outcome > 5 ) && (bs == 1))
            advice = sprintf('%s (long) has risen more than 5 percent above
predicted value, cash in now!',s)
    elseif((outcome > 2 ) && (bs == 1))
            advice = sprintf('%s (long) has risen more than 2 percent above
predicted value, consider cashing in',s)
    end
end




end
```